NEW RADIAL BASIS FUNCTION NETWORK BASED TECHNIQUES FOR HOLISTIC RECOGNITION OF FACIAL EXPRESSIONS

DE SILVA CHATHURA RANJAN

MEng. (Nanyang Technological University)

B. Sc. (Computer Science and Engineering), University of Moratuwa)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgement

I wish to express my sincere appreciation and gratitude to my supervisors, Dr. Liyanage C. De Silva and Dr. S. Ranganath for their guidance and encouragement extended to me during the course of this research. I am greatly indebted to them for their time and efforts spent with me over the past four years in analyzing problems that I have faced through the research. I would like to thank Dr. Ashraf Kassim for all the assistance given to me during my stay at the National University of Singapore.

I owe my thanks to Ms. Serene Oe, Mr. Henry Tan and Mr. Raghu, from Communications Lab and Multimedia Research Lab for their help and assistance. Thanks are also extended to all my lab mates for creating an excellent working environment and a great social environment.

Success of my research program may not have been reality without the invaluable supports form my wife, Nayanthara and my family. I would like to appreciate their encouragements, patience and support extended to me during the four year of this research. A special thank goes to my brother Dr. Harsha De Silva for all his advice on the medical and surgical aspects of the human facial anatomy.

I would like to thank the management and staff at the Dept. of Computer Science and Engineering, University of Moratuwa for allowing me for an extended stay at the National University of Singapore in order to complete my research programme.

Lastly but not the least, I would like to thank all my friends and colleagues who kindly agreed to be test subjects in the facial image database. My sincere gratitude is extended to Dr. Jeffrey Cohn of Carnegie Mellon University for providing his facial expression image

database for my research work. A special thank goes to my friends Sarath, Upali and Malitha for their assistance given printing this thesis.

Table of Contents

Summary

With a number of emerging new applications, automatic recognition of facial expressions is a research area of current interest. However, in spite of the contributions that have been made by several researchers in the past three decades, a system capable of performing the task as accurately as humans remains a challenge. A majority of systems developed to date use techniques based on parametric feature models of the human face and expressions. Because of the difficulties in extracting features from facial images, these systems are difficult to use in fully automated applications. Furthermore, the development of a feature model that holds across different cultures and age groups of people is also an extremely difficult task.

Holistic approaches to facial expression recognition on the other hand use an approach that is more similar to that used by humans. In these methods, the facial image itself is used as the input without subjecting it to any explicit feature extraction. This entails using classifiers with capabilities different from those used in parametric feature based approaches. Typically, classifiers used in holistic approaches must be able to handle high-dimensionality of the input, presence of irrelevant information in the input, features that are not equally important for separation of all the pattern classes and the ability to learn from a small training data set.

This thesis focuses on the development of Radial Basis Function (RBF) network based classifiers, which are suitable for the holistic recognition of expressions from static facial images. In the development, two new types of basis functions, namely, the Differentially Weighted Radial Basis Function (DWRRBF) and the Cloud Basis Function (CBF) are proposed. The new basis functions are carefully crafted to yield best performance by using

the specific properties of the problem domain. The DWRRBF use differential weights to emphasize differences in features that are useful for the discrimination of facial expressions, while the CBF adds an additional level of non-linearity to the RBF network, by segmenting basis function boundaries into different arcs and using different radii for each segment to best separate it from its neighbors. Additionally, by using a combination of algorithmic and statistical techniques, an integrated training algorithm that determines all parameters of the neural network using a small set of sample data has also been proposed.

The proposed system was evaluated and compared with other schemes that have been proposed for the same classification problem. A normalized database of static facial images of test subjects from a range of cultural backgrounds and demographical origins was compiled for test purposes. The performance of the proposed classifiers and several other classification methods were tested and evaluated using this database.

The proposed RBF network based classifiers demonstrated superior performance compared with traditional RBF networks as well as with those based on popular dimensionality reduction techniques. The best overall recognition rates of 96.10% and 92.70% were obtained for the proposed CBF network and DWRRBF network classifiers, respectively. In contrast, the best performance among all other types of classification schemes tested using the same database was only 89.78%.

List of Symbols and Nomenclature

Unless stated specifically the following context of symbols and nomenclature are used throughout this thesis.

$\mathrm{var}(x)$	Variance operator of variable x
Σ	Covariance matrix
Σ_j	Class conditional covariance matrix of class j
$\boldsymbol{\mu}$	A column vector of mean data
$\boldsymbol{\mu}_j$	Mean vector of class (cluster) j
$\mathbf{W}_{pca}$	Principal Component (PCA) projection matrix
$\mathbf{W}_{fld}$	Fisher's Linear Discriminant (FLD) projection matrix
$\mathbf{S}_B$	Between class scatter matrix
$\mathbf{S}_W$	Within class scatter matrix
$\mathbf{x}_j$	A column vector of j^{th} data input
x_{ij}	The i^{th} element of input vector $\mathbf{x}_j$
$\mathbf{y}_j$	A column vector of network output corresponding to $\mathbf{x}_j$
y_{ji}	The i^{th} element of the network output $\mathbf{y}_j$
$\mathbf{t}_j$	The target vector corresponding to input data $\mathbf{x}_j$
t_{ij}	The i^{th} element of the target vector corresponding to input data $\mathbf{x}_j$
$\phi_j(\cdot)$	The j^{th} basis function in a RBF network
$\boldsymbol{\varphi}(\mathbf{x})$	Response of basis functions in a RBF network corresponding to input $\mathbf{x}$
$\mathbf{W}$	Weight matrix

σ_j^2	Variance of j^{th} data cluster or overall radius of j^{th} basis function
U_j	Set of parameters associated with the j^{th} basis function
$\boldsymbol{\Theta}_j$	Discriminative indices of the j^{th} basis function
Θ_{ij}	The i^{th} Discriminative Index of the j^{th} basis function
S_k	Subset of images belonging to k^{th} ths subject in the databse
C_k	Subset of images in the database labeled as expression class k
η	Learning rate

Symbols in bold type face letters are used to represent vector quantities and matrices while symbols in normal italic typeface are used to represent scalar quantities. Unless specifically stated, a column of a matrix represents a single observation whereas a row of a matrix represents a single variable.

Except in the literature survey in Chapter 2, the term "Cluster" is used to represent data in local neighborhood that may not necessarily have to the same class label. The term "Class" is used represent data with the same class label whereas the term "Homogeneous Cluster" is used to represent data in a local neighborhood and having the same class label.

List of Figures

List of Tables

CHAPTER 1

Automatic Facial Expression Recognition and Its Applications: An Introduction

In face-to-face human communication facial expressions are an integral component of the interaction. According to some psychologists, the extent of information conveyed through such paralinguistic means even surpasses the amount of information conveyed verbally. For example, a study by Mehrabian [1] revealed that as much as 55% of the information is conveyed through facial expressions, while the balance is conveyed through verbal and other non-verbal actions. Moreover, facial expressions are a means of expressing one's emotional state. Hence, recognizing facial expressions is an important component of human social interactions.

Apart from face-to-face communication, the importance of facial expressions has also been highlighted recently in human-machine interactions. With recent developments in advanced Human Computer Interfaces (HCI), researchers have pointed out that facial expressions could be used as an effective method of communication between humans and machines. An advanced User Interface (UI) with the capability of recognizing facial expressions would be able to recognize the user's emotional state and then adjust its responses accordingly. Video conferencing systems could save valuable communication channel bandwidth by recognizing and transmitting parametric descriptions of the speaker's facial expressions instead of streaming facial images. This information can then be used to reconstruct a facial image with corresponding expressions at the receiver.

Advanced HCI systems with capabilities in facial expression recognition have additional applications in field of robotics. For example, a robotic pet dog developed recently by Sony consumer electronics [2] at present is capable of only responding to voice commands and some visual cues from its user. With an embedded automatic facial expression recognition system, these robots, in the future, will be able to respond to their owner's emotions in a similar way to a live pet.

With numerous potential applications, development of automatic facial expression recognition systems is an interesting topic of current research. However, in spite of numerous contributions in the literature a system that can match a human's ability in this task is yet an open problem. Furthermore, a majority of techniques reported so far use computations that may be quite different from the way humans recognize and interpret facial expressions. For example, most approaches discriminate expressions based on different parametric models of the face. This is different from the holistic approach taken by the human brain for recognition and analysis of faces. Although some of these model-based techniques have demonstrated excellent capabilities in recognition of expressions from their model parameters, determining such parameters automatically from facial images still remains a difficult and computationally expensive task.

In this thesis, a holistic facial expression recognition system that takes a more human-like approach to solve the problem is proposed. The emphasis is placed on the development of a suitable pattern classifier for the problem, using a Radial Basis Function (RBF) neural network architecture. In the development, several enhancements to the network, including two new types of processing nodes are proposed. The test results have shown that the proposed classifier is capable of recognizing facial expressions with an accuracy of 96.10% on the test images, compared to a best of 89.78% achieved using other types of classification schemes.

1.1 Facial Expressions and Human Emotions

Emotions and facial expressions are two different but related phenomena of human behavior. From a neurological point of view, expressions that appear on the face are results of neuromuscular activities of facial muscles, triggered mostly by the emotional state. In one of the earliest published investigations in the late 1640's, John Bulwar [3] suggested that it is possible to infer the emotional state of a person from the actions of his facial muscles. A more comprehensive study on specific muscles related to emotions and facial expressions was published many years later by Duchenne [4] in the early 1860's. During these experiments, moist electrodes were attached to key motor points on the subject's face. Thereafter, small "galvanic" currents were applied to these electrodes and observations on the resultant facial articulation were recorded. From the experimental results, Duchenne was able to identify isolated muscles or small groups of muscles that were expressive of the emotional state. Accordingly, these facial muscles were even named by the author using their associated expressions, as "muscle of joy", "muscle of crying", and "muscle of lust" etc.

1.2 Universal Facial Expressions and Their Effects on Facial Images

Psychologists believe that there are six universal types of facial expressions that can be recognized across different cultures, gender and age groups [5]. These categories include expressions of "Fear", "Surprise", "Angry", "Sad", "Disgust" and "Happiness". However, within these categories there can be numerous levels of "expression intensities" with varying details that are displayed on the face. Faigin [6] described some of these details from an artist's point of view as shown in Figure 1.1. According to him, there are three main regions in the human face including the eyes, eye-brows and the mouth region which display a majority of the details in facial expressions. For example, expressions of "Fear" and "Sadness" make the inner portion of eye-brows bend upwards whereas expressions of "Anger" causes the same to bend downwards. Similarly, the eye-brow region in general

remains relaxed during expressions of Happy and Disgust but is raised during expressions of Surprise and Fear.

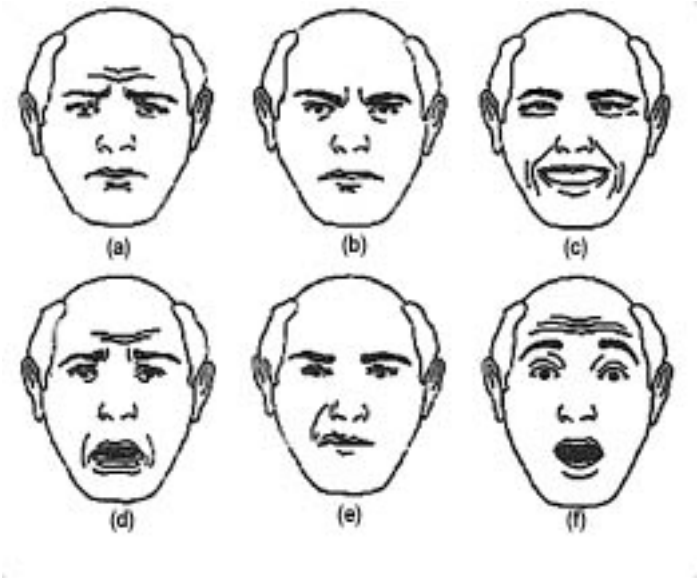


Figure 1.1: An Artist's point of view of the six universal classes of facial expressions [7]. (a) Sad, (b) Angry, (c) Happy, (d) Fear, (e) Disgust and (f) Surprise.

The shape of the eyes during a facial expression is determined by the pressure applied on the lower eye lids by the upper cheek region and on the upper eye lids by the eye brows. Lack of such pressure on the eyelids makes the eyes open wide during Surprise and Anger. Similarly, the pressure from upper eyelids usually causes eyes to remain partly closed during the expression of Sadness. The mouth region of the face is most illustrative in Happy, Fear and Surprise expressions. When expressing Surprise, the mouth takes a round shape while the Happy expression makes the mouth to be open wide open with lip-corners pulled backwards. The mouth may also be wide open during extreme Fear but usually stays closed when expressing Anger and Sadness.

In addition to above, several expressions cause some transient features like wrinkles to appear. These features in general, include horizontal folds that appear across forehead and upper eyelids during expression of Sad, Fear and Surprise and those appear below the lower

lip in expression of Happy and Fear. Additionally nose-wrinkles are also common in expression of Happy, Fear and Disgust due to the upward movement of the inner cheek region.

1.3 Recording and Describing Facial Changes

Because of the subjectivity in linguistic descriptions of facial expressions and other changes in the face, researchers have developed formal techniques that can be used to record and describe facial signals more accurately and consistently. There are several versions of these techniques often used by practitioners of psychology to identify and record the subject's emotional states [8]. Among these, the Facial Action Coding System, the Maximally Discriminative Facial Movement Coding System and the MIMIC Language are widely used in psychology as well as in the description of facial signals for computer-based face analysis.

1.3.1 Facial Action Coding System and Maximally Discriminative Facial Movement Coding System

The Facial Action Coding System (FACS) [9] describes visible motion of the face in terms of primitive building blocks called Action Units (AU). Each Action Unit corresponds to a single change in the facial geometry, without any regard to facial muscle(s) causing such change. For instance, in upper face region AU1 corresponds to "inner brow raise" while AU2 correspond to "outer brow raise" (Figure 1.2). In lower face region, "upper lip raise" corresponds to AU10 whereas "jaw drop" and "mouth stretch" correspond to AU26 and AU27 respectively. The complete FACS system consists of 56 such Action Units, of which 44 account for mostly non-rigid motion of the face and changes caused by facial expressions.

(a) (b) (c)

Figure 1.2: Examples of Action Units in FACS [10]. Images of (a) AU1, (b) AU2 and (c) AU4.

It must be noted that the FACS itself is completely based on an anatomical basis of facial movements and therefore does not make any explicit references to the underlying emotions nor the facial expressions caused by such emotions. Nevertheless, as has been pointed by many researchers [11] it is possible to infer facial expressions as combinations of different FACS Action Units. The relationship of these AU's to the six universal facial expressions is described in Table 1.1.

Facial Expression	AU coded description
Happy	AU6 + AU12 + AU16 + (AU25 or AU26)
Sad	AU1 + AU4 + (AU6 or AU7) + AU15 + AU17 + (AU25 or AU26)
Anger	AU4 + AU7 +(((AU23 or AU24) with or not AU17) or (AU16 + (AU25 or AU26)) or (AU10 + AU16 + (AU25 or AU26))) with or not AU2
Disgust	((AU10 with or not AU17) or (AU9 with or not AU17)) + (AU25 or AU26)
Fear	(AU1 + AU4) + (AU5 + AU7) + AU20 + (AU25 or AU26)
Surprise	(AU1 + AU2) + (AU5 without AU7) + AU26

Table 1.1: Relationship between FACS Action Units and classes of universal facial expressions.

In contrast with the FACS system the Maximally Discriminative Facial Movement Coding System (MAX System) [12] records only a restricted set of facial movements, in terms of some preconceived categories of emotions. This technique is primarily intended for recording of emotions in infants and therefore is based on eight different categories of emotions often displayed by infants. Similar to FACS, the MAX system also records only the visible changes in the face without any regard to facial muscles acting on them.

1.3.2 The MIMIC Language

While both FACS and MAX systems were developed primarily for recording of facial signals irrespective of the facial muscles associated with them, the MIMIC language [13] on the other hand was developed for the reverse; i.e. for description of facial expressions in

terms of the muscular activities. MIMIC assumes that facial expressions are direct results of both static and dynamic aspects of the face. Static aspects are primarily based on the structural effects of facial bones and soft tissues, and therefore are not influenced by the emotional state. In contrast, dynamic aspects of the face are the direct effects of the emotional state. The MIMIC language describes the latter effects in terms of actions by "mimic muscles" in the face.

Compared with FACS and MAX systems, MIMIC language is a powerful tool in the description of facial expressions in terms of various parametric models. Consequently, this technique is widely used as a scripting tool in many facial animation systems.

1.4 Applications of Automatic Facial Expression Recognition Systems

Until recently, Automatic Facial Expression Recognition (AFER) systems were developed mainly as supporting tools for psychological practice and for human behavior analysis. These systems were expected to help in the tedious task of monitoring and recording the subject's emotional states either with on-line systems or using pre-recorded video. However with some recent developments in HCI applications and the availability of low cost CCD cameras and higher computing power, AFER systems have found their way into a number of new emerging areas of applications.

One area of application that would benefit most by AFER systems is computer-based distance learning systems. Unlike a classroom environment, instructors involved in distance learning facilities do not get direct feedback from students through eye contact. Receiving such information through live video feedback is also not realistic in most cases due to the high bandwidth requirements and the distributed audience. However, using an AFER system installed in the remote classroom, an alternative method of emotional feedback can be

constructed. For instance, feedback such as "90% of the students are confused" will allow the instructor to re-explain his material.

A similar application area that would benefit from AFER systems are Computer-Based Training (CBT) systems. These days, almost every computer has a CCD-based digital camera as one of its standard accessories. Using this device, a background process could analyze a user's facial expressions, and generate information regarding his/her emotional state to the CBT system. Thereafter depending on the emotional intensity corresponding to surprise, confusion, frustration and satisfaction etc., the CBT system can monitor the user's learning process and adjust its level of explanation to suit the user [14].

Facial expression analysis is also applicable in advanced transportation systems. A camera with an embedded AFER algorithm can monitor the alertness / drowsiness of the driver and then generate an appropriate warning when necessary. In aircraft, such a system can detect emotions related to stress/panic conditions of the pilot and alert the control tower when necessary. Additionally AFER systems could also activate safety shutdown mechanisms in hazardous machinery when their operators are detected to be sleepy or drowsy.

Research by Ekman et. al. [15][16] has discovered evidence which relates micro-facial expressions to whether someone is telling the truth. For instance, when a person is truly enjoying himself his smile is accompanied with muscular activities around the eyes whereas with fake smiles such muscle activity is not present. These observations show that AFER systems can also be used as a potential tool for lie detector tests. Moreover, unlike conventional polygraphs where "probes" have to be physically attached to the subject, an AFER based system would require only a non-invasive camera. Consequently, they can be used transparently and in real time in any environment, such as court-rooms, police investigation rooms etc. where ascertaining truthfulness is of crucial importance.

Apart from the above, AFER systems are also finding applications in a number of emerging disciplines. These include but are not restricted to computer games, software product testing, communication / linguistic training and several internet applications like chat rooms, virtual teleconferencing systems [17][18]. In general, wherever an autonomous system requires information about the emotional state of its users, AFER systems will have a significant role to play.

1.5 Motivations for this Research

For humans, analysis of facial expressions is a very simple task which is carried out hundreds of times each day with virtually with no effort. However for computers, it is a sophisticated problem that requires complex algorithms and techniques in image analysis and high dimensional pattern recognition. For this reason, in spite of the numerous contributions that have been made in the recent past, an AFER system with capabilities close to human recognition still remains an open problem.

In general, humans and computers use approaches that are quite different to each other in recognition of facial expressions. Neurological evidence has shown that human perception of faces and their expression is a holistic process involving a feed-forward neural mechanism [19]. In contrast to the human approach, a majority of the computer-based methods use some anatomical feature model of the face in order to describe and analyze facial expressions. This approach requires several geometrical and motion features parameters to be extracted from facial images which are then fit to an anatomical model. Although the classification results recorded from these approaches are convincing, they often underrate the complicated process of successfully extracting such features in an autonomous way. Furthermore, the development of a universal anatomical model for faces across different cultures, age groups and demographical origins is also a difficult task at best.

Hence, there has been growing interest in the development of human-like approaches to AFER systems. These approaches process and recognize facial images holistically without any explicit extraction of anatomical or motion parameters from them. However, due to the absence of a parametric anatomical model these approaches often require the ability to work with high-dimensional feature vectors and typically adopt a connectionist framework to discriminate between classes of facial expressions. Although the results that have been recorded so far are less convincing than their model-based counterparts, these systems offer a number of benefits. For instance, they can be highly adaptive and learn through examples without any a priori knowledge of an underlying parametric model. Furthermore, classifiers like RBF neural networks have additional advantages such as fast learning algorithms and the ability to work with wide variations in the input [20] and these offer several benefits to AFER systems. Additionally, their low processing power requirements and adaptable properties often makes them ideal candidates for implementation in embedded systems.

An holistic approach to AFER typically consists of two major components. The first acquires and segments the facial image from its background, followed by normalization for variations such as camera scaling, translation, rotation and differences in intensity. The second component on the other hand is a classification system that discriminates facial expressions using the normalized image. While several advanced image processing and analysis techniques are available for the first task, significant improvement is still required for the second with respect to specific aspects of AFER systems. In this thesis some of these improvements, using a platform based on Radial Basis Function networks are investigated.

1.6 Major Contributions of this Thesis

A novel approach for classifying facial expressions holistically from facial images is developed. Using the RBF network architecture as the basis, a new classifier capable of recognizing facial expressions without any explicit extraction of feature parameters or the

use of a priori knowledge of anatomical features of facial images is developed. In the development of the proposed classifier, the following major contributions have been made.

- An extensive investigation of the current and past methods of AFER systems has been carried out to find the advantages and disadvantages of various approaches to the problem. Additionally, some of these methods were implemented to obtain bench-mark results using the same data set used in the evaluation of the proposed methods. An extensive study of practical problems encountered in designing RBF network classifiers for high-dimensional spaces has also been carried out.

- Two new types of basis functions for RBF networks have been developed. These basis functions were designed to incorporate the capability of learning local properties of the problem domain in high-dimensions with fewer training data compared to existing types of basis functions. Furthermore, they have also been tailored to address specific problems in holistic approaches to AFER systems, such as the presence of irrelevant variations due to subject's identity information etc. in the input.

- An algorithmic approach for designing the classifier and a new criterion based on the Raleigh coefficient [21] for initializing the new basis function parameters have been proposed. These algorithms use an iterative procedure to determine the minimum number of basis functions required by the network according to the properties of the training dataset and the stipulated performance goals.

- A series of experiments have been done to evaluate the performance of the proposed new classifiers for recognizing facial expressions. A database of facial images belonging to test subjects of various cultures and demographical origins has been created for evaluation of different classifiers. The test results have shown superior performance of the proposed methods compared to several other types of RBF network classifiers and statistical classification methods.

1.7 Organization of the Thesis

This thesis is organized into seven chapters. In Chapter 1, the background information about automatic facial expression recognition and some of its applications are presented followed by the motivations of this research and the major contributions that have been made in this thesis. In Chapter 2, an extensive literature survey of techniques that have been used for facial expression recognition is presented. Additionally, performances of past methods are also compared against the general expectations of "an ideal facial expression recognition system". This is followed by a detailed discussion of algorithms, properties and issues in designing RBF network classifiers for high-dimensional pattern recognition in Chapter 3. Details of the development of the proposed classifiers and related algorithms are discussed in Chapter 4. A brief description of the image database used in the evaluation of the proposed classifiers is presented next in Chapter 5. In Chapter 6, classification results of the proposed classifiers are presented and discussed. In addition, the results obtained with other types of RBF network classifiers and those using common dimensionality reduction methods are also presented in Chapter 6. Finally in Chapter 7, concluding remarks of this thesis and some directions for future research are presented.

CHAPTER 2

Successes and Failures in Automatic Facial Expression Recognition: A Literature Survey

Until a few decades ago most of the literature on facial expression analysis was published by psychologists. One of the earliest works originating from a non-psychology background was Suwa et. al. [22], which described a preliminary investigation on automatic recognition of facial expressions from image sequences. Since then the topic has drawn the attention of scientists and researchers working in a variety of fields. During the last decade, automatic facial expression analysis has been of interest in areas like man-machine interfaces, computer vision and pattern recognition. As a result of this renewed interest, a number of techniques and algorithms have been proposed for computer recognition and analysis of human facial expressions. Some of the major achievements in the recent past are presented in this chapter.

2.1 Introduction

For humans, recognition of facial expressions under different conditions is an effortless task. However for computers, it is a complicated problem that requires a combination of complex algorithms and techniques from computer vision, image analysis and pattern recognition. Appearance of the face differs considerably from one individual to another due to differences in their age, gender, ethnicity, demographic origin and sometimes due to the presence of occluding objects like eye-glasses and facial hair. Moreover, faces are likely to appear under various conditions including differences in pose, lighting and in cluttered backgrounds. These variations must be addressed properly at various stages of the facial expression recognition system in order to make them usable in practical applications.

When building an automatic facial expression system, the designer must first make key decisions on three major aspects of the system. These are; (i) how the expression information is presented to the recognition system, (ii) the nature of feature extraction and (iii) the type of classifier for final categorization of expressions. Over the last two decades, researchers have proposed a range of techniques and algorithms that address various issues related to these tasks. In the following sections these developments are discussed under the broad categorization illustrated in Figure 2.1.

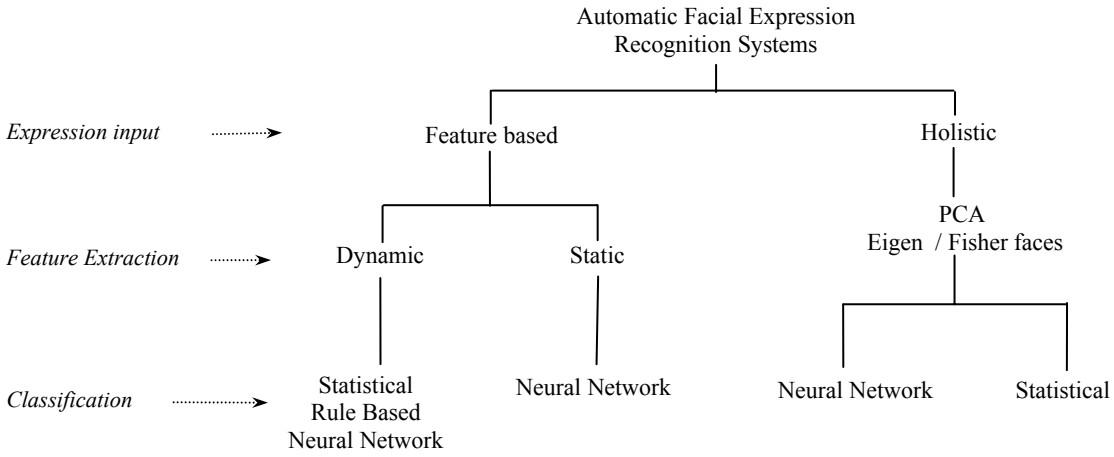


Figure 2.1: Categorization of techniques used for automatic facial expression recognition.

Since the early days there has been an ongoing debate within the research community regarding the best composition of input space for automatic recognition of facial expressions. Some researchers favour a feature-based representation [23] where information about facial expressions is presented using a set of low dimensional measurements obtained from facial images. Others favour presenting faces holistically as two-dimensional (2-D) or one-dimensional (1-D) arrays of pixel intensities [24]. In the 1-D representation, the image is often transformed onto a vector using row or column concatenation.

One of the often cited difficulties in the holistic approach is their higher dependency on external environmental conditions like lighting and background. Therefore, to minimize the effect of these dependencies on the recognition, facial images often have to be acquired

under strictly controlled conditions. In contrast to the holistic approach, measurements used in feature-based methods are chosen to provide some degree of invariance to these external factors. As a result, these systems may appear to be more robust when operating in practical environments. However, automatic detection of such invariant features in practice is again a difficult task, and reliable feature extraction remains a problem.

In feature-based methods, there are two basic types of measurements which are considered to be good descriptors of facial expressions. Some researchers suggest that dynamic non-rigid motion of the face is the best way to describe facial expressions whereas others argue that the same can be achieved through static measurements, such as those describing geometrical shape of important facial components (eyebrows, eyes, mouth etc.). Arguments supporting the suitability of these two types are often taken from a psychological view point. For instance, most psychological research on facial expressions over several decades has been successfully conducted using "mug-shot" images showing expressions at their peak level [25]. These images have been effectively used to find expression cues such as changes in the shape of the eyebrows, eyes, the mouth and the presence of transient cues like wrinkles. On the other hand some other experiments have shown that even non-rigid motion of the face with minimum spatial detail is sufficient for the identification of expressions. For example, during a series of experiments by Bassili [26], a group of human operators who had been trained for the analysis of facial expression were shown an image sequence that contained only white dots on a dark surface of a person's face displaying different expressions. The results showed that they were able to recognize all classes of expressions close to 50 per cent accuracy using the motion of these white-dots on the dark background.

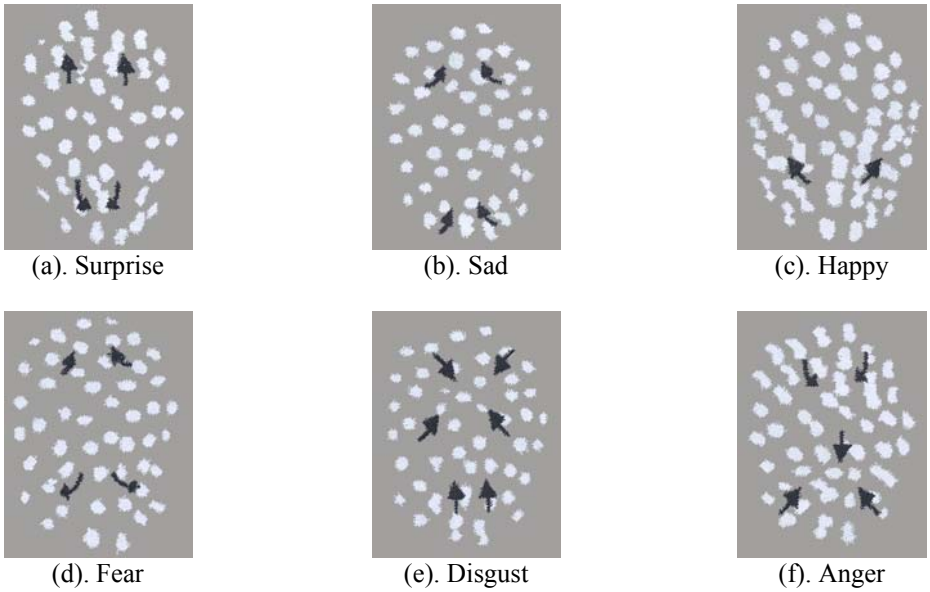
Methods based on holistic representation of the input usually do not perform any explicit feature extraction, except perhaps for dimensionality reduction. Instead, they depend more on pattern classifiers that are able to identify some intrinsic discriminative features from the noisy input. Common algorithms used in this respect include those based on principal

component analysis (PCA), Fisher's linear discriminant function (FLD) and neural networks. Feature based representations on the other hand require comparatively less complicated algorithms for classification because the features themselves are often better separable and relatively free of noise.

Although Figure 2.1 outlines categories of the most common algorithms used for automatic facial expression recognition, a clear separation of these techniques is seldom seen in practical implementations. Instead, researchers have used various combinations of available techniques and algorithms in addressing several issues related to the problem. In the following sections of this chapter some of these approaches will be discussed in detail under the three broad categories of motion-based methods, model-based methods and holistic methods.

2.2 Motion-Based Methods

Early evidence that established a relationship between non-rigid facial motion and facial expressions surfaced in Bassili's [26] experiments in the late seventies. During these experiments, with human subjects, Bassili was able to identify several principal directions of motion that were providing vital cues to observers about facial expressions (Figure 2.2). Although his observations did not associate these motion patterns with specific facial muscle actions, they provided important details about the non-rigid motions that occur during facial expressions. In addition to Bassili's experiments further evidence was also found in Ekmans's Facial Actions Coding System (FACS) [9], which described the visible changes in the face due to muscle actions. Most of the FACS Action Units (AU's) are linguistic descriptions of the movements in facial regions. For example, two of the Action Units, AU1 and AU4 are described as "outer brow raiser" and "brow lowerer", respectively.



(a). Surprise (b). Sad (c). Happy

(d). Fear (e). Disgust (f). Anger

Figure 2.2: Motion cues from Bassili's experiments [26]. Observers were shown only the motion of white patches on a dark surface of the face.

The optical flow algorithm is undoubtedly the most common technique used to extract motion details from facial image sequences. The algorithm is computationally demanding but provides a reliable estimate of the apparent motion. Optical flow in general is defined as the pixel velocities obtained from an image sequence, and arises due to the movement of brightness patterns. It can be determined using one of several techniques that could establish a correlation between pixels of a small neighborhood in two successive frames of an image sequence. For example, one such algorithm [27], which is commonly used for face processing, assumes that the brightness of an object remains constant during motion within a short time interval. This assumption constrains the image motion vectors to satisfy

$$I_x \cdot \mathbf{u} + I_y \cdot \mathbf{v} + I_t = 0 \tag{2.1}$$

where $I(x,y,t)$ is the intensity at a point (x,y) at time t, and $\mathbf{u}$ and $\mathbf{v}$ are the horizontal and vertical components of optical flow at point (x,y). In order to solve for the two unknowns in (2.1), a smoothness constraint that minimizes

$$f(\mathbf{u},\mathbf{v}) = \iint \left\{ \left(\mathbf{u}_x^2 + \mathbf{v}_y^2\right) + \left(\mathbf{u}_y^2 + \mathbf{v}_x^2\right) \right\} dxdy \tag{2.2}$$

at every (x, y) is used. Optical flow vectors $\mathbf{u}$ and $\mathbf{v}$ at point (x, y) can then be obtained by solving (2.1) with the smoothness constraint in (2.2). The optical flow solution includes motion components due to both non-rigid motion within the face as well as rigid motion of the head. Therefore it is common for many optical flow based implementations to make the restrictive assumption that the overall rigid motion of the face is negligible between any two consecutive image frames.

Further to the above basic framework, several enhanced techniques for optical flow computation [28][29][30][31] have been proposed in the recent past. Algorithms for facial motion detection use either Dense Flow Analysis (DFA) or Feature Point Tracking (FPT) described below. The primary difference between these two paradigms is the fact that the first determines motion in several regions of interest while the second focuses on the motion of only a few important feature points.

2.2.1 Dense Flow Analysis

In systems using DFA, features are computed in terms of average flow velocities over a uniform grid of small regions on the face. Typically, these regions are determined regardless of any specific facial feature or facial organ. One of the earliest applications of DFA for face processing was documented by Mase and Pentland [29], who developed an algorithm for lip-reading in facial image sequences. Afterwards, Mase extended their algorithm to facial expression recognition using a two-fold approach that was described as the 'top-down' and 'bottom-up' methods of expression recognition [30]. The top-down method suggested the creation of a face muscle model based on optical flow. This muscle model could then be related to Ekman's FACS for subsequent recognition and analysis of their facial expressions.

The bottom-up approach divided the 256×240 pixel facial image evenly into rectangular regions of 16×15 pixels in size without considering where the primary muscles of

expression interact with the facial skin. For each region in the grid, dense optical flow was first computed throughout the complete duration of an expression image sequence. Thereafter, five different parameters using first and second order moments of optical flow data in spatial and temporal domains were computed for each region. As a result, for 256 regions in the facial image a total of 1280 features were computed from the optical flow data.

In order to reduce the dimensionality of the feature space to a level manageable by the underlying classifier, the author suggested the elimination of feature variables that provided little information for discrimination of different expressions. Such direct elimination of features was feasible since all regions of the face do not have an equal participation in creating expressions in the face. In order to quantify the usefulness of each of the 1280 features, the author suggested a criterion function which estimated the goodness of each feature k as

$$j(k) = \frac{\text{var}_B(k)}{\text{var}_w(k)} \tag{2.3}$$

where $\text{var}_B(k)$ and $\text{var}_W(k)$ are the between-class and within-class variances of the k^{th} feature. Only the top 15 features, that scored highest according to (2.3) were included in the final set of features used in the classification. The final categorization of expression classes was done using a k-nearest neighbor rule criterion. The results showed a success rate of 80% on a test database consisting of 30 image sequences obtained from 10 different subjects. With the removal of eight potentially ambiguous image sequences, the recognition rate increased further to 86%. However, the scope of the database itself was limited only to four ("Happy", "Anger", "Surprise" and "Disgust") classes of facial expressions.

Later in 1994, Yacoob and Davis [31][32] proposed a system based on localized Dense Flow Analysis, that was capable of handling all six classes of universal facial expressions. For a reduced feature space, the algorithm focused only on motions associated with eye brows,

eyes and the mouth regions that are considered as the primary components of the face associated with expressions. Optical flow was computed at high gradient values in these regions. Authors also suggested thresholding and quantization of motion vectors in order to eliminate minor variations due to noise and other related factors.

The final classification of motion variables onto facial expressions was thereafter carried out using a rule-based system, which was created from a psychological background. Observations made by Bassili [26] and linguistic interpretation of FACS Action Units were used as the basis for construction of the rule base. Decision rules were applied in three temporal stages; beginning, peak and ending of an expression in order to maintain required coherence among different subjects in the temporal domain. Tests on the algorithms were conducted using a database of 105 image sequences belonging to 30 individuals. The highest recognition rate of 94% was recorded for Surprise while Anger and Disgust recorded 92% percent recognition rate. The system recognized 85% of Fear and Happy expressions while the lowest score of 80% was recorded for the Sad expression.

More recently some researchers have proposed neural network based classifiers for the categorization of facial expressions from motion parameters. In one such attempt Rosenblum et. al [33] used a Radial Basis Function (RBF) network for the classification of localized motion parameters originating from two classes (Smile and Surprise) of facial expressions. The network inputs were the Dense Flow parameters obtained using an optical flow algorithm operating at high gradient points in regions of eye brows, eyes and the mouth. After experimenting with different types of RBF networks and different network parameters, the authors recorded their best results using two categories of test images, consisting of familiar and unfamiliar test subjects. Familiar test subjects whose images were used for network training recorded recognition rates of 85% and 93% for Smile and Surprise expressions, respectively. In comparison, unfamiliar subjects whose face images were not

included in the training set recorded a slightly different recognition rate with 83% for Smile and 94% for Surprise.

In a separate investigation Masahide et. al. [34] combined DFA with a discrete Hopfield network for final classification. In this method, a normalized face image was first divided into a grid of 8×10 rectangular regions of equal size and local DFA was used to compute optical flow in these regions. Following this, each individual region was assigned to one of three discrete feature values; (+1, 0 and -1) respectively for "upward motion", "neutral" and "downward motion" based on vertical components of the averaged local dense flow. Finally these discrete feature values were used in a Hopfield neural network for the categorization into expression classes. Test results on 4 expression classes yielded individual recognition accuracies of 78% for Anger, 88% for Sadness, 99.4% for Surprise and a perfect 100% for Happiness.

In practice, facial expressions performed naturally are accompanied with a certain amount of pose variations. However, when such head movement is present, motion information becomes less descriptive of facial expressions because of the co-occurrence of rigid and non-rigid motions in the same sequence. In fact, the results shown for all motion based algorithms discussed so far were under the restrictive assumption of negligible head motion during the expression sequence. Black and Yacoob [35] addressed this problem by using a collection of parametric flow models that accounted for both rigid and non-rigid motions. The parametric models were developed using separate image flow models constructed concurrently for the entire face movement and motions in localized regions of eye-brows, eyes and the mouth that corresponded more to facial expressions. Tests were carried out with a database of 138 expression sequences from 40 different subjects. During these tests, the subjects were allowed to move their head but without creating profile views in the facial image. The results showed recognition rates ranging from 87% to a perfect 100% for different types of expressions.

Algorithms based on DFA in general depend on optical flow computed over multiple regions of the face. Therefore, when part of the face is occluded, these techniques are likely to encounter problems in representing data for subsequent classifications [36]. Techniques using optical flow of small regions of the face in contrast are likely to face fewer problems in handling occlusions. Although the occlusion causes loss of some information, parameters that are not affected by occlusion can still be used for classification. However in the latter case their underlying classifiers too must be able handle the partial input data.

2.2.2 Feature Point Tracking

In contrast with DFA, methods that use Feature Points Tracking (FPT) compute motion parameters of only a small set of prominent facial feature points. Typically, these features are related to regions like eye-brows, mouth corners and lip-boundaries etc. Compared to DFA regions, these more salient features not only reduce the risk of tracking loss but can also be detected more accurately when automatic feature detection algorithms are employed. Often these features are detected on the first frame of image sequence and are thereafter tracked through the rest of the frames using computationally simpler algorithms. As a result FPT requires less computational power than DFA where optical flow needs to be computed on all frames in the sequence. This computational advantage makes FPT more suitable for real-time applications.

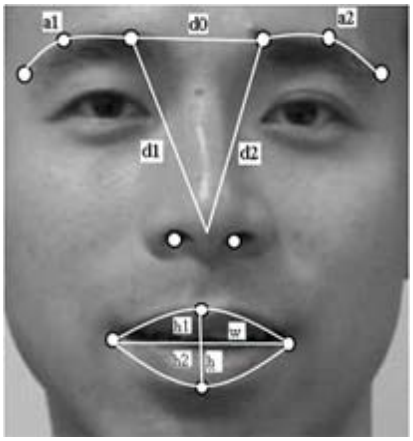
In 1995, Moses et. al. [37] developed a system that was capable tracking mouth shape in real-time. The tracker used the valley of pixel intensities that is usually visible in between upper and lower lips of the mouth region. The authors preferred valley detection over edge detection citing inconsistencies and multiple occurrences of edges during various stages of muscle actions. The valley contour was tracked using a Kalman filter [38] that used both real-time measurements as well as prediction based on an a priori model of the contour dynamics. Using this algorithm the authors were able to determine five different shapes of

the mouth which included Neutral, Smile, Sad, Open and the "OO" shape. All confusions between shapes recorded during tracking were associated only with the Neutral shape. Although the experiments were limited to the shape of the mouth, the authors suggested that the same procedure can be used for other facial features and thereby for the recognition of all types of facial expressions.

In a separate development, Otsuka et. al. [39] proposed a system that was able to model motion parameters of almost the entire face by tracking only a few feature points. The authors' main objective however was to use the motion information to determine FACS Action Units. The tracking algorithm that was built around the Kanade-Lucas-Tomasi [40] tracker was capable of locating and tracking vital feature points automatically with minimum user interventions. In the first frame of the image sequence the feature points were located using local extrema or saddle points of luminance distributions belonging to facial regions of interest. Next by using a triangulation method that eliminated geometrically redundant points, the number of features required for tracking was further reduced. Thereafter during subsequent frames, a number of motion parameters were computed by tracking these feature points. Finally, by considering muscle contractions associated with each of the triangulated feature points the FACS action units were determined.

Tracking algorithms in most cases return noisy features due to external environmental effects like changes in lighting, presence of transient features, shadows and head motion. Additionally, complete or partial loss of tracking parameters could also occur when there is occlusion in the facial image. Although the effects of noise can be compensated to a certain extent by using spatial and temporal filtering coupled with a quantization process [31][32], the effects of occlusions in image sequence are almost non-recoverable. Typically, handling of occlusion requires adaptation of a feature representation model to compensate for the loss of information [41]. Recently, Bourel et. al. [42] addressed these issues by combining feature-point tracking with a state-based representation of feature parameters. The main aim

in this approach was to compute a set of motion feature parameters that are more tolerant to noise and loss of information using a finite state transition model.



h1	:	Upper lip height
h2	:	Lower lip height
h	:	Lip height
w	:	Mouth width
d0	:	Inner brows distance
d1,d2	:	Nostrils / brows distance
a1,a2	:	Eyebrow angle

Figure 2.3: Feature points and measurements for state based representation used by Bourel et. al. [42].

The algorithm tracked 12 feature points located around the mouth, the nose tip and the eye brows (Figure 2.3). Nostrils were used as anchor points for features as well as for tracking of head motion and orientation. First, for each frame in the image sequence nine spatio-temporal measurements as illustrated on Figure 2.3 were computed using the selected feature points. Thereafter, using a state transition algorithm each measurement was converted into one of three possible states named as “increasing”, “stable” and “decreasing”. Upper and lower thresholds for state transitions were determined using the average motion amplitude of individual features concerned. Finally, the discrete feature vectors obtained by concatenating all nine states were then classified into six expression classes using a rank weighted k-nearest neighbor classifier [43]. However, in their published results [42] the authors did not state any numerical figures regarding the success rates in their approach. Nevertheless the graphical illustrations of the results (Figure. 2.4) that appeared in their publication revealed the following interesting facts.

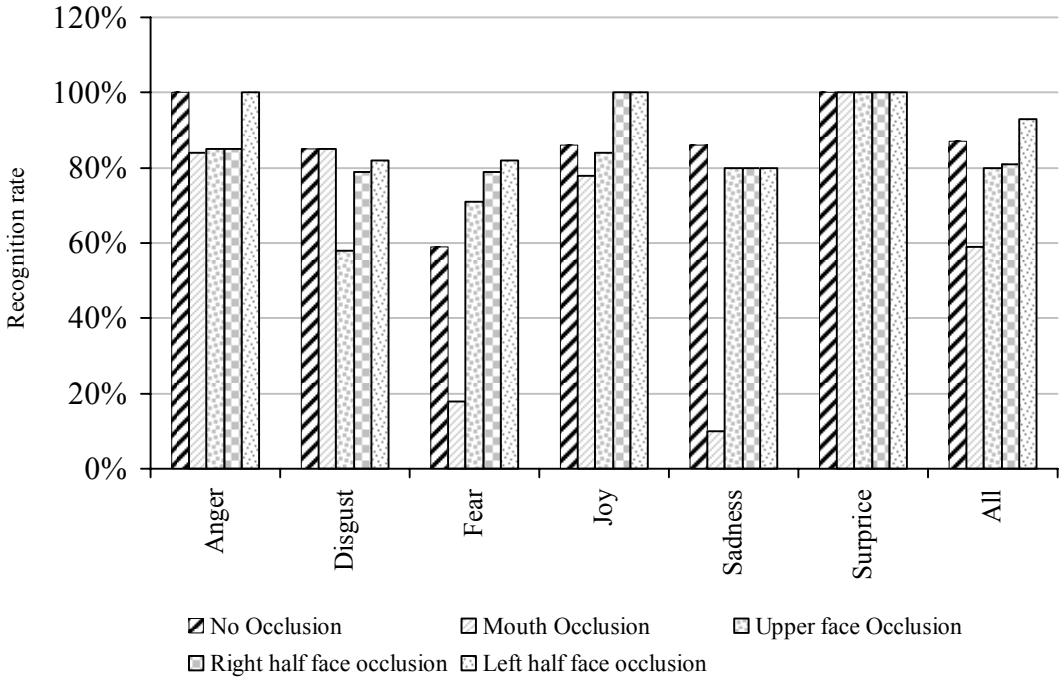


Figure 2.4: Recognition rates reported by Bourel et. Al. [42].

1. Anger and surprise expressions recorded an almost perfect recognition rate (close to 100%) while disgust, joy (happiness) and sadness recorded a rate above 90 percent. The lowest recognition rate, close to 60% was recorded for fear expression.

2. Occlusion resulted in lower recognition rate for all expression classes except for fear expression, which recorded a significant increase (from ≈ 60% to ≈ 90%) when the left half of the face was occluded.

3. Occlusion of the lower face (mouth region) created a more significant drop in the recognition rate compared to occlusion of the upper face.

4. On average left and the right halves of the face carried equal amounts of information regarding the facial expressions.

One of the useful observations made by Bourel et. al. [42] was that occlusion of either left or the right half of the face had a minimum impact on the recognition rate compared with that of full images. This suggests that facial expression can still be recognized using only half of the facial image. The reduced input size of the half-face image is likely to bring several advantages, especially for holistic representations where dimensionality is a major issue.

Compared with DFA algorithms, FPT has been applied only in a few instances for recognition of facial expressions. Nevertheless recently there have been a growing number of investigations that use these techniques for analysis and recognition of FACS Action Units [41][44][45]. Although the objectives of such investigations were not directly aimed at recognition of facial expressions, their results; i.e., the FACS Action Units can still provide a firm basis for the same task.

2.3 Model-Based Methods

While motion-based techniques use dynamics of facial muscles as the main source of feature extraction, model-based techniques in contrast use static information that appear as the end-effect of such muscle actions. Consequently, they represent faces and facial expressions in terms of distance measures, angles or their moment-invariants with respect to facial components like eyes, eye-brows, nose, mouth and the chin area. These measurements are typically computed either from static facial images or sometimes from video data with image frames captured at discrete intervals. Consequently, feature extraction for model-based methods requires less computations compared to motion extraction, where motion vectors must be computed or tracked for each and every frame in the sequence.

Features computed from static images can be used in two different ways. They can either be classified directly into facial expressions or can be interpreted by fitting them into some anatomical model of the face. Accuracy of the first approach depends more or less on the

individual feature detection algorithms used to compute the features. In contrast, the accuracy of the second approach depends on the validity of the model in describing facial expressions. However, in both the approaches building an invariant deterministic model of the human face for all expression classes is a difficult task because of the variety of structural differences that exists across individual faces.

One of the earliest investigators to use static features was Kobayashi and Hara [46] who developed a facial expression recognition system that was based only on measurements taken from static facial images. The measurements were related to a set of 30 Facial Characteristic Points (FCP) originating from three facial components: eye brows, eyes and the mouth region (Figure 2.5). These facial components were selected based on Ekman's Facial Action Coding Systems (FACS) [9] considering the fact that 39 out of 44 Action Units (AU's) were directly associated with the facial organs concerned. Thereafter using FCPs in these regions some sixty different distance measurements were computed relative to corresponding FCPs on a facial image with neutral expression.

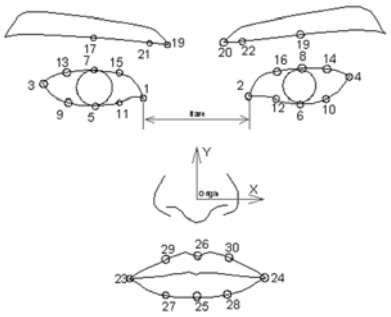


Figure 2.5: Facial Characteristic Points (FCP) used by Kobayashi and Hara [46].

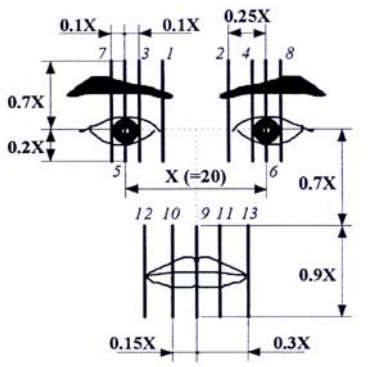


Figure 2.6: Position of vertical lines for scanning for facial features [47].

Translation invariance of features was obtained by taking all measurements using a normalized coordinate system that originated from the centre of the nose tip. Moreover,

prior to computing such measurements the FCPs themselves were normalized in order to account for camera scaling and other structural differences in individual faces. For this purpose the authors selected a line connecting two inner corners of the eyes (between feature points 1, 2 in Figure 2.5), which in general was found to be invariant to facial expressions. All feature positions were then normalized based on the length and the angle of this line with respect to the horizontal direction. Finally, the categorization of feature measurements into expression classes was made using a neural network classifier. The network consisted of four layers; an input layer with 60 neurons, two hidden layers with 100 neurons each and an output layer with 6 neurons corresponding to each of the six facial expression classes. The neural network was trained using a supervised gradient descent procedure with the error back propagation algorithm.

In order to test their algorithms, the authors created a database of Facial Characteristic Points from 172 facial images belonging to 30 different subjects. All the feature points were manually coded off-line by a human operator. Out of the 30 subjects, data belonging to 15 were used for training the neural network. Results obtained using the remaining 15 subjects showed recognition rates of 86.7% for Surprise, 91.7% for Fear, 92.3% for Anger, 92.9% for Sad and 84.6% for both Disgust and Happy expressions. The overall recognition rate was recorded as 88.7%.

During a later investigation the authors further developed their facial expression recognition system with real-time recognition capabilities [47]. Compared to the previous work, the new system had the capability of detecting FCPs automatically and then classifying them into facial expressions. For the localization and detection of feature points the following procedure was used. First eye-brows and the pupils were searched within the image using intensity profiles along vertical lines. This was possible because the colour of eye brows and the pupils made a clear contrast from the surrounding facial skin. Next, the rest of the features were detected by scanning along 13 different vertical line segments in the regions of

eyes and the mouth. Positions of these lines were determined empirically and relative to the distance between the centers of the two pupils (Figure 2.6).

Deviating from their first attempt which used measurements taken from 30 discrete feature points, the new approach used intensity distributions along the 13 vertical scan lines on the face as the input to the neural network classifier (Figure 2.6). In order to compensate for the differences in the size of each individual face, the facial images were first normalized using an affine transform such that the distance between the two pupil centers was 20 pixels. After normalization the final feature vector obtained for classification consisted of 234 (18 * 13) pixel values. The neural network classifier had 3 processing layers with 234 neurons in the input, 50 neurons in the hidden layer and 6 neurons in the output layer. Similar to their previous approach the network was trained using the error back-propagation algorithm with training data obtained from 15 subjects.

Tests done with the balance of 15 subjects in their image database showed results compatible with the previous design where the feature points were manually coded. Authors recorded their best recognition rates for the new system as 60% for Disgust, 80% for Anger, 100% for Happiness and 90% for Surprise, Fear and Sad expressions. The overall recognition rate was recorded as 85%.

In another development Ushida et. al. investigated the use of Conceptual Fuzzy Sets (CFS) to categorize information extracted from facial feature points [48]. Selection of CFS for the classification scheme was made with consideration to the subjectivity involved in the description of facial expressions. The classifier consisted of three Kohonen networks [49] followed by the CFS network. Each Kohonen network was trained to specialize on features extracted from one of three facial regions; namely eye brows, eyes and the mouth. The authors used the same image database as Kobayashi's experiments but restricted their

classification to three (Angry, Happy and Sad) expression classes. After training the classifier using data from 9 subjects an overall recognition rate of 78.7% was obtained.

More recently another FCP based automatic facial expression recognition system was proposed by Jyh-Yeong Chang et. al [50]. However in contrast with Kobayashi et. al.'s algorithm which used structural relationship between facial organs, the authors proposed a new technique based on the shape of eye brows, eyes and the mouth. The shape information was determined, first using Rough Contour Estimation Routine (RCER) [51]. Once the contours were determined their corresponding FCPs were estimated using a method similar to that of Kobayashi. Classification of FCPs was also tested using two different neural network architectures, a Multi-Layer Perceptron (MLP) network and a Radial Basis Function (RBF) network. Both networks however, produced similar results with an overall recognition rate of 92.1%. The image database used for these experiments consisted of 80 images out of which 42 were used for network training.

A rather different approach was taken in the JANUS [52] system which attempted to reason about facial expressions from visual cues appearing on the face using a rule-based system. The JANUS system first converted face geometry into several static linguistic action formats like "brows raised", "eyes open" and "jaw dropped" etc. Thereafter, the expressions were determined by matching these actions with corresponding muscle movements associated with basic expression classes in a rule-based classifier. Comparing their results with those obtained by human experts and non-experts in facial expression analysis, authors showed that the JANUS system was capable of performing close to non-trained human operators.

Researchers at the Delft University of Technology in Netherlands developed another rule-based system called HERCULES (Human Emotion Recognition Clips Utilized Expert System) which used measurements computed from 20 different facial feature points. The measurements were first used to infer FACS Action Units and subsequently facial

expressions [53][54]. The feature points themselves were selected based on their association with the primary facial muscles of FACS Action Units and on their suitability for automatic detection. Majority of the feature points were located in the eyebrows, eyes, nose tip and the mouth regions similar to the FCPs proposed in Kobayashi's work. Categorization of feature measurements was carried using a rule-based inference scheme that made up the core of HERCULES system. The categorization included two stages. In the first stage, Action Units of the FACS system were derived from the feature measurements and in the second stage they were further classified into individual expression classes.

2.4 Holistic Methods

In contrast to feature based algorithms, holistic approaches use images as their input without performing explicit feature extractions. Often the classifiers used in these methods use a connectionist model, which consists of a network of simple processing units. The network then develops receptive fields, typically in a layered architecture to recognize *intrinsic* features that can discriminate pattern classes. A prominent advantage of connectionist style classifiers is that they do not necessarily require a priori knowledge about the organization of these receptive fields or the intrinsic feature model. Instead, these properties are generated by learning through a set of examples. The connectionist nature and learning paradigm often make these methods more capable of handling noisy, partial and even potentially conflicting data at their inputs. For recognition of facial expressions, such characteristics are highly beneficial because of the variety of structural and environmental differences in facial images of different people.

However the advantages of holistic methods are available only at the expense of some of the beneficial properties of feature-based techniques. For example, feature-based techniques are more robust under external environment conditions such as changes in the background, lighting and presence of transient features during an expression. On the other hand, holistic

approaches are more sensitive to these conditions because such changes would directly affect the representation of their image-based input. As a result holistic representations require more robust and extensive preprocessing and normalization of the input compared to their feature-based counterparts. Nevertheless, most of these normalizations can be carried out using automatic techniques that require minimum user-intervention.

Another major problem faced by holistic algorithms is the extremely large dimensionality of the input space compared to the number of unique data patterns available for classifier training. Researchers believe that the minimum spatial resolution required for simple face detection is at least 32x32 pixels [55]. However, even at this resolution the holistic input would contain over 1000 features. Additionally, because of the connectionist architecture the network may also contain a large number of learnable parameters amounting to several times the dimensionality of the input. For many learning algorithms finding a global solution for such a large number of parameters using a small data set is a difficult task at best. Added difficulties are some unavoidable variations in facial image space that are not related to expressions. For instance a facial image may contain many details about a subject's identity, gender and the age. Often these properties cause a much higher variability in the input space, compared with the useful ones caused by the facial expressions. Therefore, when trained with only few samples the classifier network becomes receptive to these more prominent variations and as a result fails to separate the intrinsic features that describe facial expressions.

To address these issues some researchers have proposed the application of dimensionality reduction techniques on holistic input. Dimensionality reduction in general has two objectives. Firstly, it can reduce the number of variables in the input by projecting data onto a possibly uncorrelated and low dimensional space. Secondly, some variables with information that is not related to facial expressions can be excluded during the projection onto the low dimensional space. The latter helps to prevent the network from learning un-

wanted details in input while the former reduces the number of features in the input to a manageable level. Jointly, both these properties contribute to improvement of the network classifier in terms of the performance and generalization.

Among the algorithms available, Principal Component Analysis (PCA) is undoubtedly the most widely used technique for dimensionality reduction. Given a set of data samples in a high dimensional space, PCA computes a set of orthogonal axes that point in directions of maximum variability of the input data. Therefore, by projecting onto these axes it is possible to obtain a set of variables in a low dimensional sub-space that retain most of the variability (energy) of the original high dimensional space. Let $\boldsymbol{\varphi}$ be a $m \times N$ matrix containing N zero mean vectors of dimension *m* along its columns. Then the covariance matrix Σ of the data sample can be computed as,

$$\Sigma = \frac{1}{N} \varphi \varphi^T . \tag{2.4}$$

Thereafter the principal component projection matrix $\mathbf{W}_{pca}$ that maximizes the variance in the projected space can be computed as

$$\mathbf{W}_{pca} = \arg\max \left| \mathbf{W}^T \Sigma \mathbf{W} \right| . \tag{2.5}$$

The columns of $\mathbf{W}_{pca}$ are the eigenvectors of the covariance matrix whereas their corresponding eigenvalues indicate the variability associated with each eigenvector. The eigenvector corresponding to the largest eigenvalue points in the direction of maximum variability while the one with second largest eigenvalue points in the direction of maximum variation and is orthogonal to the first and so on. If there are N independent data vectors in φ, eigen decomposition of the covariance matrix will contain $N-1$ non-zero eigenvalues. Therefore by selecting $N', \left(N' < N\right)$ eigenvectors corresponding to the largest non zero eigenvalues, it is possible to obtain the $m \times N'$ projection matrix, $\mathbf{W}_{pca}$ that retains the most

of the variability in the projected sub-space. Thereafter for a given data vector $\mathbf{x}$, its image projected on the principal component sub-space can be computed as

$$\mathbf{p} = \mathbf{W}_{pca}^{T}\mathbf{x} \tag{2.6}$$

where $\mathbf{p}$ is a vector with $N', \left(N' \ll m\right)$ elements.

The application of PCA to facial image analysis became popular after a pioneering investigation by Turk and Pentland [56], who used PCA for face recognition. After being motivated by a previous research that investigated PCA for image compression and reconstruction [57], the authors argued that "if a multitude of face images can be reconstructed by a weighted sum of a small collection of characteristic images, perhaps the same set of weights can also be used to characteristically represent the original images". The collections of characteristic images then represent a sub-space (face-space) that best accounts for the distribution of face images within the input image space. Since each of such face-like images in the characteristic set were the eigenvectors computed from the input image space, they were referred to as "Eigenfaces".

Using Eigenfaces, Turk and Pentland first showed that PCA can effectively be used for dimensionality reduction for face image analysis. Since then Eigenfaces and its derivatives have become the *de-facto* method for dimensionality reduction in a number of holistic face analysis systems. The recent literature shows two major trends in the application of PCA for facial expression recognition. Some researchers have proposed Eigenfaces or PCA to be applied on facial regions as the primary feature extraction method for classification while others suggested that PCA be combined with other holistic feature extraction methods.

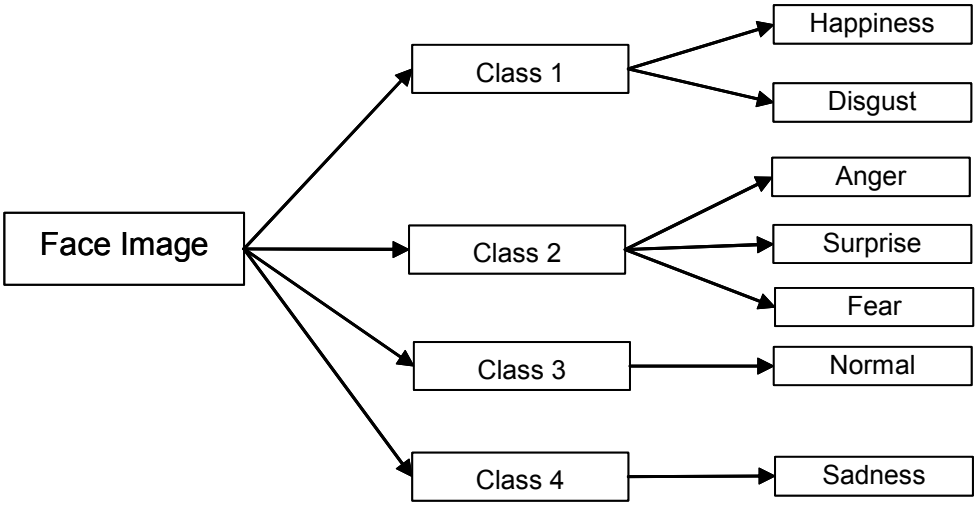


Figure 2.7: Two level classification proposed by Daw-Tung et. al [58].

Using the coefficients of PCA projections directly as features, Daw-Tung and Jam Chen [58] proposed a Radial Basis Function (RBF) network classifier that recognized seven different types of expressions on facial images. The seven types of expressions are the six universal expression types and the neutral face. The RBF network provided discrimination of the PCA sub-space using a two-level hierarchical classification process. The first level (referred by the authors as the "mouth layer") of the hierarchical structure categorized inputs into one of four intermediate classes while the second level (eye layer) further subdivided them into one of the seven expression classes (Figure 2.7). Each level of the hierarchical structure had its own PCA subspace and RBF network computed from the images belonging to their respective pattern classes. The authors supported their decision for a two-level hierarchical structure citing the relative dominance of the mouth and the eye regions in describing different facial expressions. The first level attempted to separate images based on the variation in the mouth region, which was considered as more prominent compared to the eye-region in separation of the four intermediate sub categories of expressions (Figure 2.7). At the second level, sub-categories with similar mouth shapes allowed the PCA algorithm to capture variations in the less prominent eye region.

A database consisting of 70 images from 10 different subjects was used to compute PCA and then train the RBF network. The authors used three different versions of the same dataset according to different levels of holistic representations as follows.

- Type I: Full face image including hair, shoulders and background. Image size was 145x175 pixels.

- Type II: Images with hair, shoulders and background removed from type I images. Images size was 80x80 pixels.

- Type III: Separate Image segments of eyes and the mouth regions. Image of sizes were 80x20 and 45x30 respectively for the two regions. The mouth segments were used to train the first level of the network while the eye segments were used for the second level.

For the analysis of network performance, another set of 70 images from a different set of subjects was used. Simulation results showed the highest recognition rate of 72.22% for type II images while type I images recorded the lowest rate of 14.28%. For type III images, the mouth region recorded a perfect recognition at the first level, but the second level which used the eye region for discrimination performed rather poorly recognizing only 57.30 % of the test images.

In a separate development Padgett et. al. [59][60] proposed the use of a Multi-Layer Perceptron (MLP) network for the discrimination of principal component sub-space computed using static facial images. Facial images were presented to the network classifier as projections of seven 32×32 pixel blocks onto a principal component sub-space of 15 dimensions. The seven pixel blocks included three originating from the right eye region and

four originating from the mouth region (Figure 2.8). However in contrast with other PCA based methods, the principal component sub-space itself was not computed using the same set of data. Instead the authors decided to compute PCA from a large ensemble of 32×32 pixel regions, selected from random locations within the facial images. Based on the results of their previous works [61], authors claimed that this representation of PCA is likely to be better in generalization compared to an eigenface / feature strategy where the sub-space is computed over the same regions as the input.

Figure 2.8: Facial feature regions used by Padgett et. al. [59].

The input to the classifier was a 105-dimensional vector computed using projections of each of the 7 pixel blocks onto the top 15 eigen vectors in the principal component sub-space. The classifier consisted of an ensemble of 11 different fully connected feed-forward MLP neural networks. Each network in turn contained an input layer of 105 nodes, a single hidden layer of 10 nodes and an output layer of 7 nodes. All eleven networks were trained independently using an on-line back propagation algorithm and with test/response pattern created from an image database. Because of the rather small image database of only 12 subjects, the classifier performance was tested using the leave-one-out cross-validation method. The network was then trained using images belonging to 11 individuals while holding the 12^{th} person as the test subject. In order to avoid the impact of a bad hold-out set, each of the 11 individuals was also used as hold-outs in turn. Finally the results obtained for all 11 different networks were combined to get an average recognition rate of 86%.

Apart from acting directly on facial images, PCA has also been used to reduce the dimensionality of feature spaces computed using other types of holistic feature extraction methods. For instance, Mathew et. al. [62] used PCA to reduce the dimensionality of a 41,740 dimensional vector obtained from the responses of a Gabor jet lattice [63] operating on a set of 240x320 pixels gray scale images. Projection of this large input-space onto a principal component sub-space spanned by a set of 35 significant eigenvectors enabled the classification to be carried by a relatively simple neural network. The network contained only a single layer with six neurons. After training the network with some precautions to avoid overtraining, the authors recorded an overall recognition rate of 85.9%.

One of the major drawbacks of PCA is that its projection matrix attempts to capture the maximum variance across the entire image set regardless of the relevance or otherwise of such variations to the underlying classification. As a result, PCA may also retain unwanted variations that may not necessarily be discriminative of intended class structure of the classification problem. When the input consists of prominent variation other than those related to the classification problem, using eigenvectors corresponding to larger eigenvalues could even degrade the overall performance and generalization due to their influence on the underlying classifiers. When recognizing facial expressions, this problem is more significant, since for a given set of face images variations due to subject's identity, lighting and background conditions are likely to be much more significant than those due to facial expressions. According to some researchers, the effect of lighting however can be minimized by discarding the first three principal components [64]. However, for variations caused by the subject's identity such a simple solution would be a difficult task. The appropriate number of principal components, responsible for such variations would depend on the nature of the training image set. Furthermore, it is yet unlikely that the first few components will only contain the unwanted variations and hence the removal such principal components will also cause the loss of some useful information.

The effects of subject's identity on PCA were clearly evident in the results recorded by Daw-Tung and Jam Chen [58]. When full face images were used, the classifier performance degraded significantly compared to using image segments that contained only the facial regions vital for facial expressions. This lower performance with full facial images can be directly attributed to the presence of person specific structural information, especially in the outer-face regions like the hair line and chin borders.

Using "Fisherfaces", a derivative of the popular "Fisher's Linear Discriminant (FLD)" function, Belhumeur et. al. [64] addressed the above problem of PCA for facial images. Although the main objective of the authors work was to develop a lighting-invariant face recognition system, they suggested that similar methods can also be used for invariant recognition of facial expressions. Unlike PCA, Fisher's Linear Discriminant (FLD) function [65] uses class-specific projections that attempt to maximize the discrimination of a given class structure in the low-dimensional space. Suppose that for a given set of vectorized image data, the between-class scatter matrix $\mathbf{S_B}$ and the within-class scatter matrix $\mathbf{S_W}$ are defined as

$$\mathbf{S}_B = \sum_{i=1}^{C} N_i \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}\right)^T \tag{2.7}$$

and

$$\mathbf{S}_W = \sum_{i=1}^{C} \sum_{\mathbf{x}_k \in C_i} \left(\mathbf{x}_k - \boldsymbol{\mu}_i\right)\left(\mathbf{x}_k - \boldsymbol{\mu}_i\right)^T \tag{2.8}$$

respectively, where $\boldsymbol{\mu}$ is the global mean, $\boldsymbol{\mu}_i$ is the mean of class C_i, N_i is the number of image samples in class C_i and C is the number of classes. Let $\mathbf{W}_{fld}$ be a matrix that projects the data onto a low dimensional subspace. For maximum discrimination in the low dimensional space, $\mathbf{W}_{fld}$ must create homogenous data clusters that are compact and well separated from each other. Therefore, using the determinants of $\mathbf{S}_B$ and $\mathbf{S}_W$ respectively as measures of the compactness and separation of data pattern classes, Fisher suggested that

$\mathbf{W}_{fld}$ be computed to maximize the ratio between the two matrices in the projected subspace as follows.

$$\mathbf{W}_{fld} = \arg\max_{\mathbf{w}} \left| \frac{\mathbf{W}^T \mathbf{S}_B \mathbf{W}}{\mathbf{W}^T \mathbf{S}_W \mathbf{W}} \right| \tag{2.9}$$

Typically $\mathbf{W}_{fld}$ can be computed by solving the eigenvalue problem,

$$\left(\mathbf{S}_W^{-1} \mathbf{S}_B\right) \mathbf{w}_i = \lambda_i \mathbf{w}_i \ , \quad i = 1, 2, \cdots, C-1 \tag{2.10}$$

where $\mathbf{W}_{fld} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \mathbf{w}_3 \ \cdots \ \mathbf{w}_{C-1}]$ are eigenvectors obtained by solving equation (2.10). However with only a limited number of training images, $\mathbf{S}_W$ can be singular and therefore a solution does not exist. To address this problem of singularity, Belhumeur et. al. [64] suggested that images be first projected onto an intermediate low dimensional space, obtained by PCA. Therefore equation (2.9) would then become,

$$\mathbf{W}_{fld} = \arg\max_{\mathbf{w}} \left| \frac{\mathbf{W}^T \mathbf{W}_{pca}^T \mathbf{S}_B \mathbf{W}_{pca} \mathbf{W}}{\mathbf{W}^T \mathbf{W}_{pca}^T \mathbf{S}_W \mathbf{W}_{pca} \mathbf{W}} \right| \tag{2.11}$$

where $\mathbf{W}_{pca}$ is computed in the usual way using the covariance matrix of all images.

Belhumeur et. al. successfully applied (2.11) to create a projection space, using which they were able to recognize faces under varying lighting conditions. Furthermore, they also did another successful experiment to determine whether a person is wearing eye-glasses. For this experiment, the images were divided into two classes as those with eye-glasses and those without glasses. Both experiments lead to recognition rates that were better than that for PCA on the same image database. Based on the results of these experiments, the authors speculated that the same criterion can be used to recognizing facial expressions by labeling the images appropriately, according to the expression classes.

Typically the PCA algorithm concentrates most of the variations in input over a few components in the principal component subspace. However, according to biological

considerations some researchers suggest the opposite: that a neural network performs better when the variability is distributed across many input neurons [66]. One example of such a neural network is the Self-Organizing Map (SOM) [20]. Motivated by these facts and the advantages of modularity in solving similar problems, Franco and Treves [67] suggested the use of a modular network architecture using a SOM for dimensionality reduction of the input.

Figure 2.9: 24x8 pixel feature region and expressions used by Franco and Treves [67].

The neural network contained four modular layers having neurons with sigmoid transfer functions. In the first layer there were 192 neurons creating the receptive field for the 24x8 pixel facial image segment that the authors used as the feature region for facial expression (Figure 2.9). Output from each neuron in the input layer was then connected through a single Hebbian [20] weight to 48 neurons in the second layer creating the SOM network. The first two layers of the network were trained using an unsupervised learning algorithm. In the third layer, the network was divided into three modular sub-networks, each specializing in one of the expression classes that was available in the training image database. Output layer of the network contained only three neurons (representing expressions classes of Happy, Sad and Surprise), each receiving input from their respective modular network in the previous layer.

Performance of the classifier was tested using static images belonging to 14 subjects. Test results showed a generalization error of 15.4%, which was lower than the 16.7% recorded for

PCA using the same image database. The individual error rates for the three classes of expressions were recorded as 7.5% for Happy, 4.4% for Sad and 5.3% for Surprise.

In other work, Katoh and Fukui [68] proposed another facial expression recognition system based on a SOM network. In contrast with the approach taken by Franco and Treves, the authors used a much larger feature area that covered most of the facial image. However, the input was divided into three sub regions as eyebrows, eyes and the mouth and thereafter each sub region was trained with its own SOM network. The input required a sequence of several image frames since the expression classes were determined by monitoring the change of winning neuron number associated with the three SOM networks during the occurrence of the expression. Details of the image database that was used for training and subsequent evaluation of the network nor any numerical figures of the performance achieved were not included in authors publication. However, in the discussion they speculated about the large amount of confusion between Sad and Disgust expression classes. This was attributed by the authors to the ineffectiveness of the test subjects in expressing the two emotions as a parallel test carried with human subjects on the same set of images also recorded similar confusion between the two classes.

Apart from techniques that used dimensionality reduction, there are other developments which attempted to determine facial expressions directly from the holistic image representations. In one such attempt Lisetti and Rumelhart [69] proposed a three layer MLP network that operated directly on 2-D pixel intensity arrays belonging to cropped facial images. The network consisted of three layers with the input layer having a single node for each of the image pixels. The hidden layer consisted of 40 nodes receiving their inputs from all neurons in the input layer. The network was trained to produce either a number in the range from 1 to 6 or from 1 to 8 respectively on the single neuron output, depending on whether the network was intended to recognize expression classes or the expression intensity.

The authors did several experiments using cropped facial images that included the eye-brows, eyes and mouth regions as well as using partial image segments containing only the lower face area. In all cases the network learned the training image sample with minimum training errors but failed to generalize with respect to subjects and images that were not included in the training set. Nevertheless, a better generalization was still observed for partial images of the lower face compared to the full facial images.

Inooka et. al. [70] proposed a unique five layered "hour-glass"-shaped MLP network architecture that was able to learn to represent facial expressions on a two dimensional feature space. The network had 10,000 nodes each at the input and output layers corresponding to an image area of 100×100 pixels. Second and the fourth layers had 25 neurons each, while the third layer was restricted to two nodes creating an hour-glass shape for the network. Using a technique, which the authors called as "identity mapping", the network was trained by presenting the same image as input as well as the target; i.e. when an image is presented at the input the network was trained to reproduce the same image at the output layer. Weight update in the intermediate layer was then carried out using the error back-propagation algorithm. After training their network with five types of facial expressions (Neutral, Happy, Surprise, Sad and Anger), the authors observed a unique representation for each of the expression classes on the output of the two nodes at the 3rd layer of the network. Consequently, this layer was named as the "emotion" layer. Furthermore, it was also possible to reproduce images of different facial expressions at the output by forcing the output of the emotion layer. Although the authors did not extend their network for subject-invariant expression recognition, their results yielded some of the early evidence on capabilities of connectionist architectures to represent facial expression in a low dimensional space.

In another development, Antonio et. al [71] proposed a Bayesian probabilistic framework that supported both face recognition and facial expression recognition. Both classifications

used maximum likelihood decision procedures. The face was divided into 4 feature regions from which 9 different types of features were computed. All feature regions ware detected and tracked automatically from video. The feature space was modeled using multivariate Gaussian distributions on a principal component subspace. Using a database of 18 subject and 6 expressions, authors recorded the maximum recognition rate of 93.2% for Happy expression. The lowest performance on the other hand was recorded for Disgust expression with a recognition rate of 79.5%.

2.5 Applications of Facial Expression Recognition: The Past, The Present and The Future

The scope of a complete automatic facial expression recognition system includes three major tasks which must be carried out automatically or with a minimum amount of user-intervention. First, before any facial expression can be analyzed, the presence of the face(s) within a scene must be detected. Second, the system must be able to locate and extract features (or feature regions) that describe facial expressions invariant to the differences in faces among different individuals and the operating environments. Typically, this step will involve segmentation of the facial image from the scene background. Third, a discriminant function must categorize the feature-space into expression classes.

For most research in facial expression recognition, facial images are assumed to be captured under controlled conditions. Usually the images are frontal views of the face on a uniform background with controlled lighting and normalized for translation, rotation and scaling. Furthermore, many algorithms may impose additional restrictions like absence of rigid motion, subjects to be free of facial hair and not wearing eye-glasses. These conditions however are seldom true in real-life situations. For most practical applications, faces will have to be detected on cluttered backgrounds under various lighting conditions and shadows. Position of face and the camera scaling are usually unknown. Moreover in some

applications, for instance in a class-room environment, the scene is likely to contain multiple faces and possibly occlusions. Nevertheless, it must also be noted that even under controlled environments, localizing a face in a digital image is not an easy task.

Item No.	Property
1	Automatic facial image acquisition
2	Subjects of any age, ethnicity and outlook
3	Deals with variation in lightning
4	Deals with partially occluded faces
5	No special markers / make-up required
6	Deals with rigid head motion
7	Automatic face detection
8	Automatic facial expression data extraction
9	Deals with inaccurate facial expression data
10	Automatic facial expression classification
11	Distinguishes all possible expressions
12	Deals with unilateral facial changes
13	Obeys anatomical rules [72]
14	Distinguishes all 44 facial actions [72]
15	Quantifies facial action codes
16	number of interpretation categories is unlimited
17	Features adaptive learning facility
18	Assigns quantified interpretation labels
19	Assigns multiple interpretation labels
20	Features real-time processing

Table 2.1: Properties of an ideal facial expression analysis system.

The scope of feature extraction depends mainly on the operating environment and the requirements of the underlying discriminant function. Feature-based methods, including those based on feature point tracking in general, require high resolution images with clearly defined facial components like eyes, eye-brows and the mouth region. Therefore, such techniques often insist on the absence of facial hair and eye glasses. Techniques based on dense optical flow on the other hand are able to cope with medium resolution images but impose restrictions on the non-rigid motion of the head during an expression. Both these conditions however are unlikely to be satisfied completely in many practical applications. Holistic approaches, in contrast, place fewer burdens on feature extraction but have their

own limitations. For instance, they are more sensitive to scaling and translation as well as differences in lighting conditions compared to feature-based methods. Moreover since they learn the categorization by example, issues related to generalization also become more significant.

We are able to recognize facial expressions in a scene virtually with minimum or no effort. Humans are capable in analyzing facial expression with little information, for instance from static images even with some occlusion. In a recent literature survey by Pantic and Rothkrantz [73] several properties that could make an "ideal facial expression system" to perform closer to their human counterparts were identified (Table 2.1). From the list of 20 different properties, the first 13 were expected in general from any facial expression analyzer while the rest were more related to specialized applications such as behavioral science research and Human Computer Interface (HCI) applications. However, among the 27 developments surveyed, authors found that none of the systems are able to satisfy all 13 general properties. While almost all the systems were able to satisfy some of the properties (for instance items numbers 5, 10,12 and 13 listed in Table 2.1), a majority of them were lacking some of the practically important properties. These included the ability to work with occlusion and in different lighting conditions. However compared to the early developments, a majority of the recent works are capable of automatic detection of faces from the scene background. Contributing factors to these improvements include the availability of new imaging technologies and higher processing power which made complex face detection and feature detection algorithms more feasible.

In the recent past, there have been a growing number of developments that use neural networks and other connectionist style classifiers for facial expression recognition. Two of the most influential factors for this renewed interest are the recent advances in neural network-based algorithms and availability of cheap CPU memory. Additionally, the use of these methods was also supported by some of the recent neurological findings regarding

human abilities to discriminate among faces. For instance, it has been shown that complex visual processing related to face discrimination is a rapid task that can be completed in approximately one tenth of a second, suggesting the involvement of a feed-forward neural mechanism[19][74].

Connectionist architectures deliver additional benefits when facial analysis procedures are to be implemented as embedded hardware systems. A connectionist style classifier in general consists of a network of small and simple processing units connected in a layered architecture. Often the processing at local units is limited to a weighted summation and a threshold function that can be easily implemented in terms of a lookup table. Consequently, they are suitable for synthesis using custom VLSI hardware or reconfigurable devices such as Field Programmable Gate Arrays (FGPA) [75][76][76]. For instance a structure of a large fine-grain FPGA can be mapped directly into that of a MLP network. Such custom-built hardware integrated with CCD image sensors and on-chip image processing functions would then provide a solid platform for facial analysis systems embedded in domestic or commercial appliances.

2.6 Summary

Above, some of the previous developments in automatic facial expression recognition systems were discussed. A couple of decades ago, facial expression analysis was a topic of interest only to practioners of psychology. In fact, most of the earlier developments in automatic facial expression analysis systems were intended only for psychological research and practices. However, with recent advances in Human Computer Interfaces (HCI), they have found their way into a variety of day-to-day applications. In the near future, a facial expression analysis system would be an essential component in any device or communication system where emotions are an important means of information transfer. For example, they will be incorporated with Internet chat programs, multi-media mobile phones,

personal digital asistants etc. These applications would require facial expression recognition systems that are simple, computationally viable and insensitive to operating environment conditions. However, most of the systems developed so far have not been able to satisfy many of these requirements.

Since the early days, there have been two different camps of thought regarding the best way to represent facial images for recognition of facial expressions. Some researchers believe that expressions are best described by a set of low dimensional feature measurements while others suggest using holistic representations as arrays of pixel intensities or images. Features used in the first type of representation can either be motion-based or model-based. Motion-based features use dynamics of facial muscle actions extracted from video to describe expressions. In contrast, for model-based methods, feature parameters are computed from static images as geometrical parameters describing the shape of facial components like eye-brows, eyes and the mouth. Classification is then made by relating the shape of these facial components with their models belonging to different classes of expressions.

The features extracted from facial images are low-dimensional, relatively separable and insensitive to the operating environment conditions. As a result, these methods require only rather simple discriminant functions to classify expressions classes. However, it must be noted that automatic extraction of these feature is a complicated process, and therefore requires some of the most complex image analysis algorithms as well as a significant amount of processing power.

In contrast to feature-based methods, algorithms using holistic representation of the input do not require any explicit extraction of feature parameters except for some normalization of the input images with respect to operating environment and lighting conditions, registration and camera scaling. However, holistic representations lead to high dimensional input spaces that often contain a significant amount of unwanted details and variations. While some

researchers proposed the use of dimensionality reduction schemes to address these issues, some others advise using connectionist-style classifiers that are capable of operating with high-dimensional input. Connectionist-style classifiers consist of a number of simple processing units arranged in a layered architecture. One prominent advantage of connectionist classifiers is their ability to learn by examples and often without any prior knowledge of the data distributions. Additionally, some of them can be trained to work with partial or potentially conflicting datasets.

One problem in using neural network classifiers for facial expression recognition is the practical difficulty in obtaining a training dataset which would be large enough to train the network for a generalized solution. Due to the larger number of learnable parameters, generalization and performance of a neural network classifier greatly depends on the amount of distinct samples used for training. Because of these reasons, researchers often prefer the use of network architectures with fewer parameters. In the following chapter details and design issues related to one such network with several properties suitable for high dimensional classification are presented.

CHAPTER 3

Radial Basis Function Networks for Classification in High Dimensional Spaces: Theory and Practice

Connectionist approaches to automatic facial expression recognition with holistic face representations require discriminant functions that are capable of operating in high-dimensional spaces. Radial Basis Function (RBF) networks [20] are often the preferred choice for such tasks due to a number of attractive properties compared to other neural networks. Some of these properties include the relatively low number of learnable parameters in the network, availability of fast training algorithms and the ability to deal with high dimensional input. Moreover, in contrast to other neural networks such as Multi-Layer Perceptrons (MLP), RBF networks can be related closely to statistical classification counterparts. In the following sections several techniques and issues related to designing and training RBF networks for high dimensional classification are discussed.

3.1 Introduction

Algorithms similar to RBF networks were first introduced as a solution for the exact interpolation problem [78]. These problems require a mapping from the input space $\{X \in \mathbb{R}\}$ onto the output space $\{Y \in \mathbb{R}\}$ subject to the constraint that a given sample of data points must be mapped exactly onto their target values as

$$f(x_i) = y_i \tag{3.1}$$

where $\{x_i \in X, y_i \in Y\}_{i=1}^{N}$ are the N pairs of input and target values in the training data set. The RBF algorithm provides this mapping through a set of N non-linear basis functions where the i^{th} basis function, $\phi_i(\cdot)$ is centered on the i^{th} data point x_i and responds to inputs in a local neighborhood of x_i. The response depends on $\|x - x_i\|$, usually taken as the Euclidean distance between input x and the centre of the basis function, x_i. The final result of the mapping is then expressed as a linear combination of the responses from all N basis functions as

$$y = \sum_{i=1}^{N} w_i \phi_i \left(\|x - x_i\| \right) \tag{3.2}$$

for $\{x \in X,\ y \in Y\}$ where $\{w_i\}_{i=1}^{N}$ are a set of weights. Using all data points in the training dataset, the mapping in (3.2) can be conveniently expressed in matrix form as

$$\mathbf{y} = \mathbf{\Phi w} \tag{3.3}$$

and

$$\mathbf{w} = \mathbf{\Phi}^{-1}\mathbf{y} \tag{3.4}$$

where $\mathbf{y} = [y_1, \cdots, y_N \in Y]^T$, $\mathbf{w} = [w_1, \cdots, w_N]^T$ and $\mathbf{\Phi}$ is a $N \times N$ matrix having elements $\phi_{ij} = \phi\left(\|x_j - x_i\| \right)$ for $\mathbf{x} = [x_1, \cdots, x_N]^T$. It has been shown by Micchelli [79] that for a large choice of basis functions, the matrix $\mathbf{\Phi}$ is non-singular provided that data points in the training set are distinct. Hence weight vector $\mathbf{w}$ can be computed using (3.4) in a single step. Furthermore both theoretical and empirical studies have shown that in the context of exact interpolation, many properties of the interpolating function are relatively insensitive to the exact form of the non-linear basis functions $\phi_{ij}(\cdot)$.

The exact interpolation constraint in (3.1) however can introduce some undesirable properties into the behavior of the network. For instance, it can cause the network output to

oscillate when noise is present in the input. Therefore, the constraints of exact interpolation in (3.1) are replaced with several refinements as follows [80][81]:

1. The number of basis functions (h) need not necessarily be equal to the number of data points (N) in the data set. Depending on the number of data samples and their local distributions in input space, h can often be selected to be much smaller than N.

2. Centres of basis functions are no longer constrained to lie on training data points, and can be determined as part of the training procedure.

3. Parameters of basis functions are determined to be compatible with the local distribution of training data and may not necessarily be the same for all basis functions. Furthermore, a bias is included in the linear mapping to accommodate the DC component of the local data distribution.

4. The dimensionality of the input $\left\{X \in \mathbb{R}^d\right\}$ and the output $\left\{Y \in \mathbb{R}^q\right\}$ spaces can be different.

With the above mentioned modifications, the general form of a RBF network can be expressed as

$$y_i = \sum_{j=1}^{h} w_{ij}\phi_j\left(\mathbf{x}, \left\{\mathbf{u}_j\right\}\right) + b_i, \qquad i = 1, \cdots, q \tag{3.5}$$

where y_i is the i^{th} variable in the output space, b_i is the bias, w_{ij} is a linear weight connecting the response of the j^{th} basis function to the i^{th} output node and $\{\mathbf{u}_j\}$ are the parameters of the j^{th} basis function. The bias parameter b_i can in fact, be absorbed into the summation by writing

$$b_i = w_{i0}\phi_0(\mathbf{x}) \tag{3.6}$$

where w_{i0} is an additional weight and

$$\phi_0(\mathbf{x}) = 1. \tag{3.7}$$

Substituting (3.6) in (3.5) provides a more compact notation for the RBF network-mapping which can now be expressed as

$$y_i = \sum_{j=0}^{h} w_{ij}\phi_j(\mathbf{x}, \{u_j\}), \quad i = 1, \cdots, q \tag{3.8}$$

A graphical illustration of the structure of a general RBF network according to (3.8) is shown in Figure 3.1. When input and output are multi-dimensional, it is common to specify (3.8) in a matrix notation as

$$\mathbf{y} = \mathbf{W}\boldsymbol{\varphi}(\mathbf{x}, \{\mathbf{u}\}) \tag{3.9}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^q$ respectively are the input and output vectors and $\mathbf{W}$ is a $q \times (h+1)$ weight matrix that connects basis function responses to output nodes. The function vector $\boldsymbol{\varphi}(\mathbf{x}, \{\mathbf{u}\})$ in (3.9) is defined as

$$\boldsymbol{\varphi}(\mathbf{x}, \{\mathbf{u}\}) = \left[\phi_0, \phi_1(\mathbf{x}, \{u_1\}), \cdots\cdots, \phi_h(\mathbf{x}, \{u_h\})\right]^T. \tag{3.10}$$

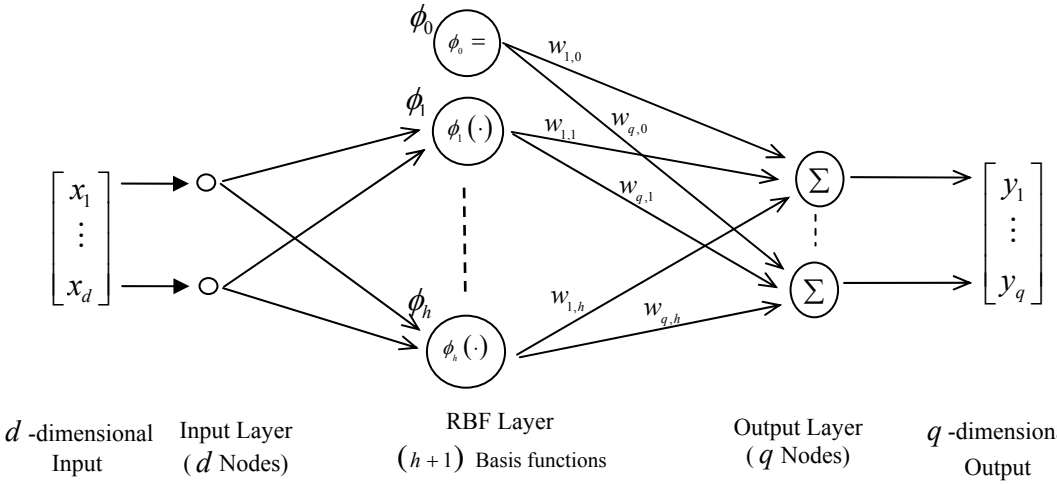


Figure 3.1: General structure of a typical RBF network.

Several different forms of radial basis functions can be used in (3.10) [82]. However, the most common type of basis function used is the Gaussian kernel,

$$\phi_j\left(\mathbf{x},\left\{\mathbf{u}_j\right\}\right)=\exp\left(-\frac{\left(\mathbf{x}-\boldsymbol{\mu}_j\right)^T\left(\mathbf{x}-\boldsymbol{\mu}_j\right)}{2\sigma_j^2}\right) \tag{3.11}$$

which is defined by the two parameters $\left\{\mathbf{u}_j\right\}=\left\{\boldsymbol{\mu}_j,\sigma_j\right\}$. Parameter $\boldsymbol{\mu}_j$ is a d-dimensional prototype vector from the input space whereas σ_j^2 is a parameter which controls smoothing effect of the kernel basis function. The response of (3.11) is maximum when the input $\mathbf{x}$ maps onto the prototype $\boldsymbol{\mu}_j$ and decays exponentially as the input deviates away from the prototype. Further normalization of Gaussian kernel (3.11) is typically not required since such constants can be absorbed into the post-basis mapping.

3.2 Properties of RBF Networks

Several properties of RBF networks make them more versatile and powerful compared to other types of neural networks such as MLP networks. RBF networks have been proven to be *universal approximators* meaning that under some mild constraints, RBF networks can be used to approximate any arbitrary function [83][84][85]. RBF networks have also been proved to have the property of *best approximation.* An approximation scheme is said to

possess this property if, in the set of approximating functions (i.e. the set of functions corresponding to all possible choices of the adjustable parameters) there is one function which has minimum approximating error for any given function to be approximated [86] . Although the former property of universal approximation can be seen in some configurations of MLP networks, the latter is more or less unique for RBF networks.

Another property that makes RBF networks different from MLP networks is the completely different roles played by the hidden layers of processing nodes in the network. A typical MLP network consists of several layers of homogeneous processing units. Each unit operates similarly to provide a weighted sum of its input followed by a non-linear activation function. On the contrary, processing units in the first (hidden) layer of a RBF network compute their responses using the "*distance"* between the input and a prototype vector at the center of the basis function. Since such prototype vectors typically are selected based on data distribution of input space, the response of the RBF hidden layer can be treated as indicative of the probabilistic relationship between an input and the data distribution. On the other hand, all nodes in a MLP network, irrespective of their topological positions within the network are part of a global distributed representation of input space. Therefore, for a given input, many of the hidden nodes will contribute to determine the output value whereas in a RBF network only the hidden nodes that are "*closer*" to the input will have a major contribution to the output value.

The different roles played by the first (basis function) and the second (output) layers of a RBF network lead to different training strategies to determine their parameters. Often, the two layers in the network are trained separately. Since nodes in the first (RBF) layer are intended to represent the data distribution of input space, their parameters are learned first using statistical properties of a large sample of un-labeled training data. Thereafter, with the RBF layer being kept fixed, weights of the output layer are determined, typically as a least square solution (3.9) using a relatively a small amount of labeled data. This property of

RBF networks is useful especially when large samples of labeled data are unavailable or when labeling large amounts of data requires a significant effort. Additionally when available, statistical information about input space can also be used to *engineer* some of the basis-function parameters in advance, thereby resulting fast training of the network.

3.3 RBF Networks for Pattern Classification

The rationale for using RBF networks in classification problems originates from Cover's theorem on the separability of patterns. The theorem states that "a complex pattern-classification problem cast nonlinearly in a high-dimensional space is more likely to be linearly separable than in a low dimensional space" [87]. Thus from a pattern classification perspective, the operation of a RBF network can be viewed as a two-tiered mapping

$$\left\{\mathbf{x} \in \mathbb{R}^d\right\} \xrightarrow{\boldsymbol{\varphi}(\cdot)} \left\{B \in \mathbb{R}^h\right\} \xrightarrow{\mathbf{W}(\cdot)} \left\{\mathbf{y} \in \mathbb{R}^q\right\} \tag{3.12}$$

where $\boldsymbol{\varphi}(\cdot)$ is a non-linear mapping from input space $\left\{\mathbf{x} \in \mathbb{R}^d\right\}$ onto an intermediate basis space $\left\{B \in \mathbb{R}^h\right\}$ and $\mathbf{W}(\cdot)$ is usually a linear mapping from basis space onto the output space $\left\{\mathbf{y} \in \mathbb{R}^q\right\}$. Typically the basis space $\{B\}$ is selected to be of higher dimension than the input $\{\mathbf{x}\}$ in accordance with Cover's theorem $(h \gg d)$.

The non-linear mapping, $\boldsymbol{\varphi}(\cdot)$ onto the basis space is a vector of basis functions

$$\boldsymbol{\varphi}(\mathbf{x}) = \left[\boldsymbol{\varphi}_1(\mathbf{x}), \boldsymbol{\varphi}_2(\mathbf{x}), \cdots, \boldsymbol{\varphi}_h(\mathbf{x})\right]^T \tag{3.13}$$

that are designed according to data distribution in the input space. The mapping can be viewed as partitioning of input space into several hyper-regions, where each region is represented by a single basis function. The response of a given basis function is maximum when a test input is at the centre of the corresponding region and decreases exponentially when the input moves away from the center. Therefore, when the partitioning consists of a set of disjoint and non-overlapping regions the response of the basis functions in (3.13)

assumes some properties of orthogonality because for any change of input within a local region, the change in the response would be restricted to a single basis function and thus only on a single dimension (axis) in the basis space. Such disjoint partitions therefore make the basis space in (3.12) to be more separable than the input space. Consequently the post-basis mapping $\mathbf{w}(\cdot)$ requires only a simple linear discriminant function in order to separate respective pattern classes in the output space.

From a statistical pattern classification point of view, it can be shown that RBF networks are closely related to Bayes' classifier [88]. Suppose that input data from the k^{th} class, $\mathbf{x} \in c_k$ can be modeled by a conditional density function $p(\mathbf{x} \mid c_k)$. Then from Bayes' theorem, the posterior probability of $\mathbf{x}$ belonging to class c_k can be obtained as

$$P(c_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c_k) P(c_k)}{p(\mathbf{x})}, \quad k = 1, 2, \cdots, K . \tag{3.14}$$

where $P(c_k)$ is the probability of class c_k and K is the number of classes. In many practical pattern classification problems, class conditional distributions of data are unlikely to be uni-modal. We assume that $p(\mathbf{x} \mid c_k)$ can be represented by a mixture of H different modes. The corresponding RBF network can be created by using H non-overlapping basis functions. Further, assuming that the H modes span the whole space of inputs, the class-conditional density function $p(\mathbf{x} \mid c_k)$ can be expressed as follows:

$$p(\mathbf{x} \mid c_k) = \sum_{j=1}^{H} p(\mathbf{x} \mid m_j) P(m_j \mid c_k) \tag{3.15}$$

where the association of density $p(\mathbf{x} \mid m_j)$ to class c_k is determined by the probability $P(m_j \mid c_k)$.

Thereafter substituting expressions (3.15) into Bayes' relationship (3.14) the posterior probability of the class membership can be obtained as,

$$P(c_k \mid \mathbf{x}) = \sum_{j=1}^{H} \frac{p(\mathbf{x} \mid m_j) P(m_j \mid c_k) P(c_k)}{p(\mathbf{x})} . \tag{3.16}$$

Introducing a unit factor $1 = P(m_j) / P(m_j)$ and comparing with (3.8) it can be shown that $P(c_k \mid \mathbf{x}) = \sum_{j=1}^{H} \frac{p($

$$= \sum_{j=1}^{H} \frac{P(m_j \mid c_k) P(c_k)}{P(m_j)} \cdot \frac{p(\mathbf{x} \mid m_j) P(m_j)}{p(\mathbf{x})} \tag{3.18}$$

$$= \sum_{j=1}^{H} P(c_k \mid m_j) \cdot P(m_j \mid \mathbf{x}) \tag{3.19}$$

$$= \sum_{j=1}^{H} w_{kj} \phi_j(\mathbf{x}) \tag{3.20}$$

represents the output of RBF network (3.8) with first and second levels of mappings being defined by

$$\begin{aligned} \phi_j(\mathbf{x}) &= \frac{p(\mathbf{x} \mid m_j) P(m_j)}{p(\mathbf{x})} \\ &= P(m_j \mid \mathbf{x}) \end{aligned} \tag{3.21}$$

and

$$\begin{aligned} w_{kj} &= \frac{P(m_j \mid c_k) P(c_k)}{P(m_j)} \\ &= P(c_k \mid m_j) \end{aligned} \tag{3.22}$$

respectively.

The posterior probability in (3.21) suggests that responses of basis functions in a RBF network can be interpreted as posterior probabilities of a given input within the hyper-region represented by the respective basis function. According to (3.22) the weights in the post-basis mapping can be interpreted as posterior probabilities of class memberships of a given input within the hyper-region of a basis function. Furthermore it can also be stated that the response of the first mapping depends on the association of input with basis functions rather than the class labels assigned to them whereas the post-basis mapping is responsible for

associating basis function outputs with their respective class labels. Owing to the above it is common to see that several techniques used to determine basis functions in a RBF network depend on large samples of unlabeled data as opposed to labeled samples.

3.4 Designing and Training RBF Networks for Classification

The design and training of a RBF network for a particular pattern classification task in general involves two major steps:

1. Determining the number of basis functions H and learning their parameters for specifying the non-linear mapping $\boldsymbol{\varphi}(\cdot)$ in the RBF layer.
2. Learning the linear weights $\mathbf{w}(\cdot)$ for the post-basis mapping.

The relationship described in (3.21) indicates that parameters of the basis functions should be determined according to the local distribution of training data in input space. However the use of only local statistics may not be sufficient for finding the parameters. Typically this is the case for types of multi-class problems where all variations in the input are not descriptive with regard to all pattern classes in the intended class structure of the pattern recognition problem. For instance, a variable which is important to separate two classes may play little role in the discrimination of a third class. Yet the local statistics of the data distribution (of the third class) will still be influenced by such variables since all basis functions must operate on the same set of input variables. Furthermore training data are also likely to contain some variations that play little role in discrimination of the intended class structure of the classification problem. Therefore if class-labels are not considered for the computation of local statistics of the input, it will not be possible to distinguish between the relevant and the irrelevant variables in determining parameters of the basis functions. Presence of these irrelevant variations will influence the "*distance*" metrics used in basis functions, possibly causing partitions of input space represented by them to become heterogeneous with respect to the intended class structure. However, it must be noted that the above problems will be less significant if the Mahalanobis distance with full covariance

matrix is used instead of the Euclidean distance (3.11) in computing the basis function responses. Nevertheless as discussed later, in high-dimensional space such an option may not be practical due to the need of an even larger amount of training samples to determine a non-singular full covariance matrix.

3.4.1 Basis Function from Subsets of Data Points

A typical Gaussian radial basis function (3.11) is defined by two parameters, the basis center $\boldsymbol{\mu}_j$ and the spread factor σ_j. Often two different strategies are used to determine these two types of parameters. The simplest method used for the first is to select a random subset of input vectors from training data as centers of different basis functions. This technique however does not provides an optimal solution with regard to the local distribution of training data and as a result could require a large number of basis functions to achieve a given performance goal. Nevertheless, the procedure is often used as a starting point for other iterative techniques that can deliver a more reasonable (smaller) number of basis functions to satisfy the performance goal. For instance, one iterative procedure starts with a large number of basis functions and then selectively removes those that cause the minimum increase in training error [89][90]. Alternatively, another algorithm can start with a small number of basis functions and iteratively add new functions until the required performance is met. In the latter case, typically in each iteration the input vector which leads to the greatest reduction in training error is retained as the new basis function.

In contrast to techniques used to determine centers, spread parameters of basis functions in general are determined more heuristically. One common heuristic approach is to set all σ_j to be equal and given by some multiple of the average distance between the basis function centers. This approach can therefore allow some degree of overlap between different basis functions to provide the much needed smooth representation of data distribution for applications such as function approximation and interpolation. For classification problems

on the other hand it is more common to use a localized approach by selecting the average distances of only k -nearest basis centers where k is usually a small integer.

3.4.2 Iterative Addition of Basis Function

This algorithm takes an incremental approach by starting with a single radial basis function and thereafter adds new basis functions until the required performance goals are met. In the first round all N training vectors are treated as potential candidates for a basis center in the single basis function network. In order to determine the best choice, N different single hidden node RBF networks are created by using each of the N training data vectors as the basis center. Each of these N networks is then evaluated and the one that provides the minimum residual error is retained as first version of the network. In the second iteration, a two node network is created by adding the best out of the remaining $N-1$ data vectors as the center of the newly added basis function. Similar to the first iteration, $N-1$ different two-node networks are created and evaluated to find the one with the minimum residual error. The procedure is then repeated, and the algorithm is stopped once the network has accumulated sufficient number of basis functions to satisfy the stipulated performance goal.

The iterative basis function addition procedure is computationally demanding because each addition of a new basis function requires a large number of networks to be created and evaluated. Hence when there are significant amounts of training data the process become less attractive due to long processing times and CPU memory required. Nevertheless, this algorithm requires almost no intervention from the user and therefore is widely used in software tools that support prototyping of RBF networks [92].

A least squares-based procedure that achieves the same goals as the above method and is more computationally efficient algorithm was proposed by Chen et. al. [93][94]. The main improvement in this algorithm is the creation of a set of orthogonal vectors in a sub-space

spanned by basis functions obtained from data in the training set. Thereafter, instead of evaluating separate networks for all input data, the data vectors leading to the minimum sum of squared error is computed directly using this orthogonal subspace. Additionally weights in second-level mapping are also computed at the same time. However, if the algorithm is continued long enough all data points in the training set will ultimately get selected as candidates for basis centers. Therefore, in order to preserve required generalization, iterations must be stopped pre-maturely when a specified performance goal is reached. A detailed discussion of the generalization issues of this least square algorithm is available in [95].

3.4.3 Basis Functions from Clustering Algorithms

As described in Section 3.3 it can be shown that the response of a basis function is related to the posterior probability of the input originating from a given local region in the input space. Consequently, use of clustering algorithms to compute basis function parameters have been proposed even when RBF networks were first introduced. For example, in one of the early developments Moody and Darken [81] used a k-means clustering algorithm [96][97] to partition the training dataset into k disjoint subsets where each cluster was thereafter represented with a basis function in the network.

The k-means algorithm in general requires the number of clusters, k to be specified a priori. Thereafter, with a fixed value of k, and given N training data vectors $\{\mathbf{x}_i\}_{i=1}^{N}$, the partitioning is determined by minimizing the sum-of-squares clustering function

$$J(S) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} \left\| \mathbf{x} - \boldsymbol{\mu}_j \right\| \tag{3.23}$$

where $\{S_j\}_{j=1}^{k}$ are subsets of training data that make up the k disjoint clusters [98]. The mean vector $\boldsymbol{\mu}_j$ of each cluster is computed as

$$\boldsymbol{\mu}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in S_j} \mathbf{x} \tag{3.24}$$

where n_j is the number of samples included in cluster S_j. Usually, the clustering algorithm starts with k randomly selected training vectors as an initial estimate for cluster centers and proceeds as follows:

1. Assign each of the k randomly selected training samples to one cluster mean such that $\boldsymbol{\mu}_j = \mathbf{x}_i$ for $i \in \{1, \cdots, N\}$, $j = \{1, \cdots, k\}$. Create a set of k clusters $\{S_j\}_{j=1}^{k}$ and assign $\boldsymbol{\mu}_j$ to S_j.

2. For each sample $\mathbf{x}_i$ in training data set where $i = 1, \cdots, N$ find the nearest cluster mean $\boldsymbol{\mu}_j$ such that $\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \leq \|\mathbf{x}_i - \boldsymbol{\mu}_l\|^2$ for all $j, l \in \{1, \cdots, k; j \neq l\}$. Then assign $\mathbf{x}_i$ to S_j.

3. For each cluster $\{S_j\}_{j=1}^{k}$ re-compute their cluster means $\{\boldsymbol{\mu}_j\}_{j=1}^{k}$ using (3.24) and the current assignment of data samples.

4. Compute the clustering function according to (3.23). Stop if the change in the criterion function $J(S)$ is below a pre-determined threshold or the cluster assignments have not been changed since the previous iteration.

5. Go to step 2.

Alternatively in step 1, the initial estimates for cluster means are sometimes computed by first assigning training samples randomly to a set of k clusters and thereafter computing their means as in (3.24). Furthermore, it has been shown that after every iteration, values of the criterion function $J(S)$ (3.23) will decrease and eventually converge when cluster assignments in step 2 become consistent [99]. However, the cluster assignment in k-means algorithm does not guarantee the global minimum of (3.23). In general, the final outcome depends on the initial estimate of cluster centers.

When dealing with large quantities of training data the k-means algorithm in the form illustrated above is likely to become expensive in terms of processing time and complexity. The number of iterations before convergence can range from a few to a few thousand iterations depending on the nature of training data and the initial estimates of cluster centers. Furthermore in every iteration, distances of all N training samples must be computed from all k cluster means. However, the number of changes in cluster-means and assignments will tend to decrease rapidly after a few iterations as the algorithm starts to converge. Exploiting this property of the k-means algorithm, a derivative known as the P-CLUSTER algorithm [100] attempts to improve the computational efficiency by reducing the number of distance comparisons involved. Using simple heuristics the algorithm keeps track of changes in means and their association with training data vectors. Distance calculations are thereby restricted only to those cluster means that have changed and data samples that have been reassigned during the previous iteration. Furthermore, the efficiency of the P-CLUSTER algorithm is also improved by the fact that as the algorithm starts to converge, movements in cluster-centers will tend to become even smaller on successive iterations.

From a different point of view, the number of distance comparisons can also be reduced by arranging data into a more efficient data structure that simplifies the problem of searching for the nearest mean [101][102]. However in contrast to some other algorithms that depend on

similar distance comparisons, clusters in k-means algorithm are dynamic and, therefore, their means are likely to change from one iteration to another. Alsabti et. al. [103] proposed to solve this problem by arranging potential candidates for mean vectors in an inverted tree structure. The tree is constructed using training data samples and therefore represents static distances between them rather than the dynamic cluster means. At the root level of the tree, potential candidates include all N samples in the training data set. On the next level, some children of the root node are pruned, based on distance properties between them. The process is repeated until each child at the last level has only a single candidate data sample. This hierarchical arrangement allows the distance computations to be restricted only within a sub-tree for most cases and thereby reduces the overall distance computations.

Connectionist-style alternatives to clustering algorithms are the Kohonen's Self Organizing feature Maps (SOM) [49]. The SOM training algorithm is an unsupervised procedure that attempts to re-arrange the placement of nodes in a low dimensional, topology preserving grid according to some similarity of patterns in the training data set. After convergence, prototype vectors in the feature map, corresponding to adjacent nodes represent neighboring regions in the input space. The SOM technique is widely used for visualization of the distributions in high-dimensional spaces. However with large dimensions, unless the problem is intrinsically low dimensional, the SOM algorithm can lead to a sub-optimal representation because of topological constraints of low-dimensional feature maps and the vast amount of dimensionality reduction involved.

One major benefit of using clustering algorithms is that the clusters can often be used to determine the spread parameter of each basis function in addition to basis centers. For instance, the sub-set of data assigned to a particular cluster can be used to compute a better estimate of the distance that separates the cluster from its neighbours in the input space. Moreover, when sufficient amounts of data samples are available, a non-singular estimate of

the full covariance matrix of the data cluster can also be made. As described later, the latter will always provide a better representation of the data spread in input space compared to a single spread factor (which yields a spherical basis function).

As described in Section 3.3 and definitions in (3.15) to (3.21), the problem of determining parameters of the RBF layer can be treated similar to a mixture density estimation problem. For example, if the input space can be modeled as a mixture of Gaussian densities, then the component densities could be used to define the basis functions for the RBF network. Following the definition in (3.15) a mixture model assumes that data distribution $p(\mathbf{x})$ to be represented as a linear combination of H kernels $\phi_j(\mathbf{x})$ as

$$p(\mathbf{x}) = \sum_{j=1}^{H} P(j)\phi_j(\mathbf{x}) \tag{3.25}$$

where $0 \le P(j) < 1$ are mixing coefficients that satisfy the constraint

$$\sum_{j=1}^{H} P(j) = 1 . \tag{3.26}$$

Once the components distributions in the mixture model are determined they can be used as basis functions while allowing mixing coefficients to be absorbed into the post-basis mapping. Algorithms that estimate mixture models from sample data usually determine both the mixing coefficients as well as the parameters associated with component density functions using similar procedures, thereby saving much of the computational load. Two of the most commonly used algorithms are the Maximum likelihood algorithm and the EM algorithm that are described in [104]. Nevertheless, these algorithms are more computationally demanding than the clustering algorithms. Additionally they require comparatively large amounts of training data to reliably estimate all the parameters of the component densities. If required amounts of data are unavailable, the number of learnable parameters in component densities must be constrained with assumptions.

3.4.4. Supervised Optimization of Basis Functions

Unsupervised clustering algorithms described in the previous section determine basis function parameters based only on given distribution of the data in inputs space. Therefore when training data contain noise and variations that are of little relevance to their class labels, resultant partitioning may be sub-optimal with respect to discriminating the class structure of the problem. For this reason, some researchers have suggested that parameters returned by these unsupervised algorithms be further optimized using supervised procedures.

A supervised update rule for basis parameters can be derived from the error back-propagation algorithm which is commonly used to train weights in MLP networks [20][82]. The algorithm minimizes the squared difference between network outputs and their corresponding target values using an interactive gradient descent procedure. Using the general form of RBF networks in (3.8), (3.11) and for a given pair $\{\mathbf{x}, \mathbf{t}\}$ of input $\mathbf{x}$ and its corresponding target output $\mathbf{t}$, the network error can be defined as

$$E(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{q}(y_l - t_l)^2 \tag{3.27}$$

and

$$\mathbf{y} = \mathbf{W}\boldsymbol{\varphi}(\mathbf{x}, \{U\}) \tag{3.28}$$

where y_l and t_l are the l^{th} elements in the q dimensional output and target vectors. For Gaussian basis functions (3.11), $\{U\} \equiv \{\boldsymbol{\mu}_j, \sigma_j\}_{j=1}^{h}$ represent learnable parameters of all h basis functions in the network. For a given post-basis mapping $\mathbf{W}$ (typically determined through procedures described in the following section) an iterative update rule that minimizes (3.27) can be obtained by moving $\{U\}$ a small distance along the negative gradient of $E(\mathbf{x})$ as

$$\{U\}^{\alpha+1} = \{U\}^{\alpha} - \eta\nabla E \tag{3.29}$$

where α is the iteration number and η is a small positive scalar value that controls the learning rate. Typically in a single iteration of the algorithm all the parameters of all the basis functions are updated. Hence using the respective partial derivatives in (3.29), update rules for the two parameters $\{\boldsymbol{\mu}_j, \sigma_j\}$ of the j^{th} basis function can be expressed as

$$\boldsymbol{\mu}_j^{\alpha+1} = \boldsymbol{\mu}_j^{\alpha} - \eta \frac{\partial E(\mathbf{x})}{\partial \boldsymbol{\mu}_j} \tag{3.30}$$

$$\sigma_j^{\alpha+1} = \sigma_j^{\alpha} - \eta \frac{\partial E(\mathbf{x})}{\partial \sigma_j} \tag{3.31}$$

where the respective partial derivatives are defined as

$$\frac{\partial E(\mathbf{x})}{\partial \boldsymbol{\mu}_j} = \sum_{l=1}^{q} (y_l - t_l) w_{lj} \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \frac{(\mathbf{x} - \boldsymbol{\mu}_j)}{\sigma_j^2} \tag{3.32}$$

$$\frac{\partial E(\mathbf{x})}{\partial \sigma_j} = \sum_{l=1}^{q} (y_l - t_l) w_{lj} \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{\sigma_j^3} \tag{3.33}$$

In order to avoid possible oscillatory behavior of learning parameters, often batch versions of the update rules are used. In these algorithms updates to parameters are computed by averaging their partial derivative over the entire set of N training $\{\mathbf{x}_i, \mathbf{t}_i\}_{i=1}^{N}$ pairs using (3.32) and (3.33) as

$$\frac{\partial E}{\partial \boldsymbol{\mu}_j} = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=1}^{q} (y_{li} - t_{li}) w_{lj} \exp\left(-\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \frac{(\mathbf{x}_i - \boldsymbol{\mu}_j)}{\sigma_j^2} \tag{3.34}$$

and

$$\frac{\partial E}{\partial \sigma_j} = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=1}^{q} (y_{li} - t_{li}) w_{lj} \exp\left(-\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \frac{\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2}{\sigma_j^3}. \tag{3.35}$$

In addition to the inherent danger of getting trapped in a local minimum of the error surface, the above gradient descent procedure does not guarantee that basis functions will be local in

input space after the training has converged [81]. Some of the basis functions in fact may evolve to respond to a very broad region in input space thereby compromising the generalization of the classifier. Furthermore, because of the exponential function, computation of partial derivatives may also get affected by the rounding off errors due to floating point representation of numerical values in practical implementations.

Similar to clustering algorithms, the iterative update of basis parameters is a computationally demanding process. On the other hand if initial estimates of basis function parameters are accurate, only a subset of basis functions are likely to generate a significant activation and have their parameters updated accordingly in each iteration. Identifying these functions in an efficient way [105] will therefore enable weight updates to be restricted only for those in responsive regions, thereby lowering overall computational load.

Based on the above concept, Schwenker et. al. [106] introduced a three-phase training algorithm for determining all the parameters in a RBF network. In the first two phases, the basis function centers and the post-basis mapping are initialized according to the usual steps, where the parameters of the basis functions are determined according to distribution of the training data set, and the post-basis mapping is determined as a least square solution. Thereafter in a third phase, parameters of basis functions are further "fine-tuned" using the gradient descent procedure described in (3.30) and (3.31). Additionally, in order to update the weights of the post basis mapping, another update rule is introduced as

$$\mathbf{w}^{\alpha+1} = \mathbf{w}^{\alpha} - \eta \frac{\partial E(\mathbf{x})}{\partial \mathbf{w}} \tag{3.36}$$

based on the gradient of error surface with respect to the weight vector. The third learning phase is then continued iteratively until the network converges at a minimum of the error surface.

A computationally simpler approach that creates basis functions representing homogeneous regions in input space can be constructed by combining both supervised and unsupervised rules into the same clustering algorithm [107][108]. In these algorithms, the training data set is first divided into homogeneous sub-sets using their class labels. Thereafter, using unsupervised algorithms such as k-means, each subset is further sub-divided into small clusters of the same class.

3.4.5 Learning the Post-Basis Mapping

Although unsupervised algorithms are common in determining parameters of basis functions, a majority of RBF networks use supervised techniques to determine their post-basis mappings. With the post-basis mapping (3.9) consisting of a linear combination of RBF layer responses, a single step solution for the weight matrix is often computed using a least-squares approach. Given set of N training input vectors $\{\mathbf{x}_i\}$, $i = 1, \cdots, N$ and their corresponding target vectors $\{\mathbf{t}_i\}$, $i = 1, \cdots, N$ the post-basis mapping (3.9) may be written in matrix form as

$$\mathbf{T} = \mathbf{W}\boldsymbol{\varphi} \tag{3.37}$$

where $\mathbf{T}$ is a $q \times N$ matrix with target vectors $[\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N]$ corresponding to their input vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, $\mathbf{W}$ is the $q \times (h+1)$ post-basis weight matrix and

$$\boldsymbol{\varphi} = \left[\boldsymbol{\varphi}(\mathbf{x}_1), \boldsymbol{\varphi}(\mathbf{x}_2), \cdots, \boldsymbol{\varphi}(\mathbf{x}_N)\right] \tag{3.38}$$

is a $(h+1) \times N$ matrix with basis function responses in (3.13) for each of the N input vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$. Then $\mathbf{W}$ is obtained from the standard least-squares solution as

$$\begin{aligned} \mathbf{W} &= \mathbf{T}\left(\boldsymbol{\varphi}^T\boldsymbol{\varphi}\right)^{-1}\boldsymbol{\varphi}^T \\ &= \mathbf{T}\boldsymbol{\varphi}^{\dagger} \end{aligned} \tag{3.39}$$

where

$$\boldsymbol{\varphi}^{\dagger} = \left(\boldsymbol{\varphi}^T\boldsymbol{\varphi}\right)^{-1}\boldsymbol{\varphi}^T \tag{3.40}$$

is the pseudo-inverse of responses from the RBF layer.

For RBF networks that are used in complex, multi-class classification problems the post-basis mapping is sometimes replaced by a perceptron network (typically single layer) with a non-linear activation function which operates on the weighted sum of RBF layer outputs. With this modification, the general form of the network output defined earlier in (3.8) becomes

$$\tilde{y}_i = f_a\left(y_i\right) \tag{3.41}$$

where y_i is the response from the activation function in i^{th} output node,

$$y_i = \sum_{i=0}^{h} w_{ji}\phi_i\left(\mathbf{x}, \{u_i\}\right) \tag{3.42}$$

defined according (3.8) and $f_a\left(\cdot\right)$ is the non-linear activation function at each of the output nodes. However, with a non-linear activation function the least-square solution described in (3.39) will no longer be valid to find the post-basis mapping. Instead, non-linear optimization techniques must be employed to determine the solution for the weight matrix. The most often used method for this purpose is the gradient-descent procedure in (3.36). The new partial derivatives of the error function in (3.36) with respect to each element w_{lj} in the weight matrix is computed as

$$\frac{\partial E\left(\mathbf{x}\right)}{\partial w_{jk}} = \sum_{k=1}^{q}\left(\tilde{y}_k - t_k\right) f'\left(y_k\right)\phi_j\left(\mathbf{x}\right) \tag{3.43}$$

where f_a' is the first derivative of the non-linear activation function. (3.44)

3.5 RBF Networks for Pattern Classification in High-Dimensional Spaces

The reasons stated in Cover's theorem [87] and the close relationship with Bayes' decision theory suggests that RBF networks are suitable classifiers for multi-dimensional pattern

recognition problems. However, when the dimensionality of the input space is large, some of these beneficial properties of RBF networks become unachievable due to some practical limitations. For instance, the curse of dimensionality makes it practically impossible to cast the high-dimensional input space onto a basis space of even higher dimensions. Moreover when the network consists of a large number of basis functions, limited availability of training data often makes clustering algorithms less reliable for determining basis function parameters. Nevertheless, it must be noted that even with these restrictions, RBF networks often outperform many other types of neural networks operating under similar conditions.

From a geometrical point of view, the task of basis functions in a RBF network is to partition the input space into several non overlapping hyper-regions. The two parameters associated with each basis function(3.11), $\boldsymbol{\mu}_j$ and σ_j respectively determine the localization and volume of the hyper-spherical region represented by the basis function. Since the categorization of these regions is carried out by linear discriminant functions (3.8) at the second level of the mapping, performance of the RBF network as a non-linear pattern classifier depends strongly on the separability of hyper-regions created by the respective basis functions. In general the best performance from the network can be expected when these regions represent homogeneous sub-sets of input data strictly separated according to their class labels.

Generalization capabilities of a RBF network depends considerably on the extent of total representation by all of its basis function in the sub-space corresponding to the problem domain within the input space. An input vector originating from a region in input space, which is not represented by any of the basis functions will not lead to any signification activation in the RBF layer, and hence will not be mapped correctly onto the output. Moreover, in high-dimensional space, input data in a particular classification problem is likely to lie within a sub-space of the high-dimensional input space. The effective

dimensionality of this sub-space is known as the *intrinsic dimensionality* of the classification problem. As described in Hartman et. al. [83], the number of hyper-spherical regions (3.11) required to fill such a sub-space increases exponentially with the intrinsic dimensionality of the classification problem. Therefore for many high-dimensional problems, representation of the full intrinsic dimensionally is practically impossible because of the exponentially large number of spherical basis functions required. In practical terms, a network with such a large number of basis function will not only require a large amount of storage space to store all its network parameters, but also require an even larger amount of training data to determine those parameters accurately.

Another condition that demands more basis functions to represent a problem domain occurs when the input space consists of features with high variations but which play an insignificant role in the discrimination of pattern classes. Irrelevant inputs like these are not uncommon in high-dimensional problems, especially when raw data are used as input or when classifiers are expected to work using some intrinsic features hidden among other variations. When data in high-dimensional spaces in fact are Gaussian-distributed, then their class-memberships are defined by Mahalanobis distances of hyper-elliptical basis function shapes as opposed to the Euclidean radii of spherical basis functions as in (3.11). If, for simplicity, hyper-spherical basis functions are used as in (3.11), then the basis functions with large radii will be required to include most of the Gaussian spread within the spherical region of the basis function. Consequently, a basis function with large radius may overlap with other representing a different class of data, thereby leading to heterogeneous class representation by the basis functions. If the input space consists of only major variations that are relevant to the separation of classes, then the radii can be decided by using the length of the major axis of the Gaussian spread. Under these conditions, regions belonging to different classes of data are more likely to be separated along their major axes. Consequently, using a radius compatible with the length of the major axes is unlikely to create an overlap with another region of a different class of data.

Presence of irrelevant variations on the other hand causes data clusters of some classes to stretch along directions that play little role in the discrimination between such classes. Yet, the radii of their spherical basis functions must be based on the length of discriminative axes in order avoid an overlap with regions of a different class. As a result their boundaries become shorter than the actual spread of data, and a portion of the data is left outside regions represented by their respective basis functions. Consequently, additional basis functions will be needed in order to include this data.

The illustrative example in Figure 3.2 demonstrates the above phenomenon for a dual-class problem in ~~a~~ two-dimensional input space. In Figure 3.2(a), separation of the two classes occurs more along the axis with larger variation. Hence, creating basis functions with radii based on the more discriminative axis will include most of the data cluster within circular regions of their respective basis functions. However, in Figure 3.2(b), the data distribution has higher variation along the non-discriminative axis x_2. Thus the spherical basis functions created for the data in Figure 3.2(b) leave a portion of data outside their respective boundaries. As illustrated in Figure 3.2(c), additional basis functions are required along the high variance axis to cover these regions that are left out by the main basis functions.

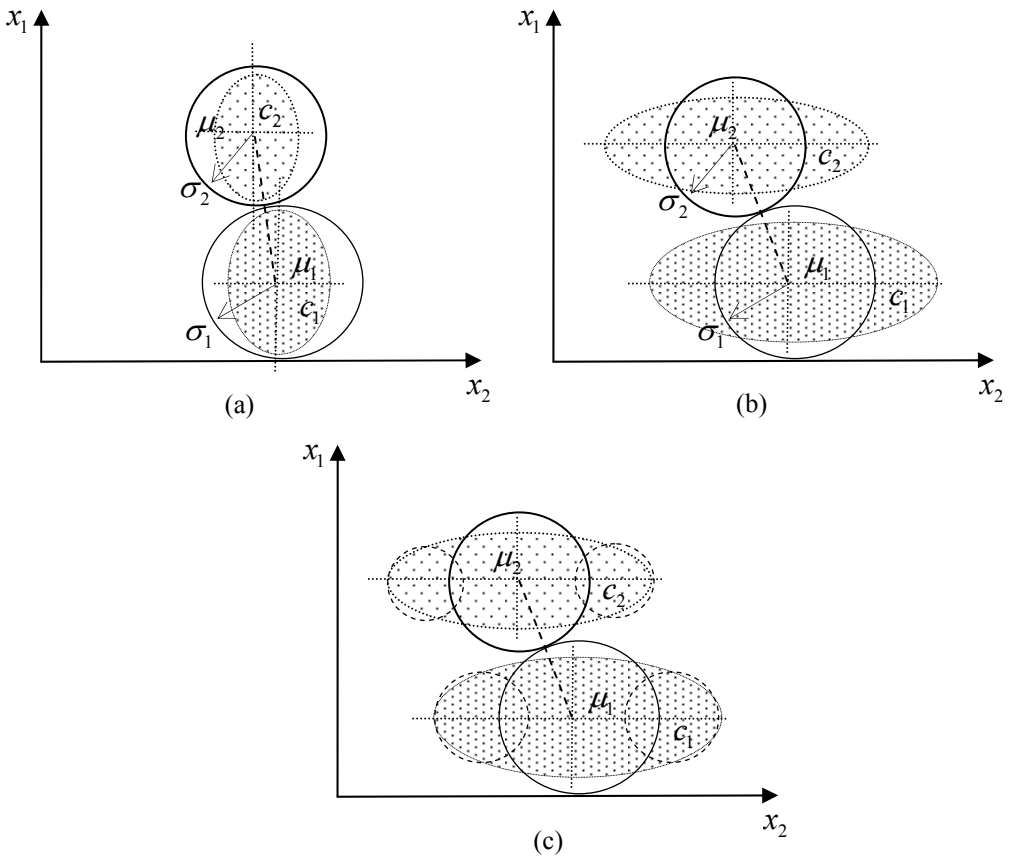


Figure 3.2: Effects of the irrelevant variables in RBF networks. (a). Discrimination occurs on the direction of major axis. (b). Irrelevant variations in x_2 variable lead to basis functions with radii shorter than the major axis of respective data spreads. (c). Additional clusters are needed to cover the spread of data.

3.5.1 An Optimal Basis Space for High Dimensional Classification

From the above discussion, it is clear that problems which arise when using RBF networks for high-dimensional classification are related to the curse of dimensionality and the presence of irrelevant variations in input space. Solutions that have been proposed to address these issues in general can be divided into local approaches and global approaches. Local approaches are based on methods to improve local representation of data by basis functions within the high-dimensional input space whereas global approaches attempt to extract important details from the input space onto some low-dimensional feature space. The latter goal in general is achieved by using explicit dimensionality reduction methods that attempt to project input onto a low-dimensional space that is optimized for the separation of pattern classes.

Dealing with irrelevant variations in input space is conceptually easier if full-covariance matrices are used to design basis functions instead of only the Euclidean radius. This defines the Gaussian basis function as

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x}-\boldsymbol{\mu}_j)}{\xi_j}\right) \tag{3.45}$$

where Σ_j is the $d \times d$ covariance matrix of data belonging to the region represented by $\phi_j(\cdot)$ and ξ_j is a positive scalar value to control the spread of the basis function response. The use of full-covariance matrix makes the Gaussian basis function better fit the data distribution of the local region compared to the spherical basis functions. However, the Gaussian basis function also contains more learnable parameters that in turn will lead to the requirement of more training samples. For instance, in order to have a non-singular estimate of Σ_j in (3.45), at least $d+1$ distinct samples are required from its local region in input space. However for most high-dimensional problems, getting such large amounts of training data is not practical.

A common workaround to the singularity issue of Σ_j in (3.45) due to rank deficiency is the use of Singular Value Decomposition (SVD) [109] to compute the pseudo inverse. The SVD decomposes the covariance matrix as

$$\Sigma_j = \mathbf{U} \tilde{\Sigma} \mathbf{V}^T \tag{3.46}$$

where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices satisfying the conditions

$$\mathbf{U}^T = \mathbf{U}^{-1}, \quad \mathbf{V}^T = \mathbf{V}^{-1} \tag{3.47}$$

and $\tilde{\Sigma}$ is a diagonal matrix that can be partitioned as

$$\tilde{\Sigma} = \left[\begin{array}{ccc|c} \omega_1 & & \mathbf{0} & \\ & \ddots & & \mathbf{0} \\ \mathbf{0} & & \omega_k & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right] \tag{3.48}$$

where $\omega_1, \cdots, \omega_k$ are the ordered k largest non-zero singular values. The total number of singular values depends on the rank of the covariance matrix which is determined by the number of training samples. However for numerical stability, in most practical cases only some a priori determined k largest values are retained. The pseudo inverse of the covariance matrix can then be computed using (3.46) as

$$\Sigma_j^{-1} \approx \Sigma_j^{\dagger} = \mathbf{V} \tilde{\Sigma}^{\dagger} \mathbf{U}^T \tag{3.49}$$

where $\tilde{\Sigma}^{\dagger}$ is defined from (3.48) as

$$\tilde{\Sigma} = \left[\begin{array}{c|c} \begin{bmatrix} \frac{1}{\omega_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{1}{\omega_k} \end{bmatrix} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]. \tag{3.50}$$

In addition to SVD and pseudo-inverse methods other statistical techniques specializing in the estimation of covariance matrices using limited amounts of training data can also be used as local approaches to optimize the basis space [110]. However, these techniques in general are computationally complex, and require substantial amounts of processing power, thus limiting their use to few applications.

It must be noted that as opposed to the true inverse of a covariance matrix, the SVD method represents only the directions of k largest variances. In order to increase the number of directions represented, some researchers have proposed the use of a common covariance matrix for all basis functions instead of those computed from their local regions [111]. In this approach basis functions differ only by their center vectors $\left(\boldsymbol{\mu}_j\right)$ and smoothing parameters $\left(\xi_j\right)$. The common pooled covariance matrix is computed over the entire training data set and therefore has a higher rank compared to its class conditional counterparts. As a result, the SVD method can better approximate the inverse due to the pooled covariance matrix computed using larger amount of data.

In contrast to local approaches, global approaches attempt to solve the problem using dimensionality reduction by explicitly projecting input space onto some low-dimensional space. The projection is made with two main objectives. First, in the low-dimensional space fewer training samples are required to obtain a reliable estimate of the parameters of individual Gaussian basis functions. Secondly, the projection can be made to retain only the relevant variables thereby reducing, if not eliminating, problems caused by irrelevant variations in the high dimensional input space.

Two of the most popular such dimensionality reduction methods, namely, the Principal Component Analysis (PCA) and Fisher's Linear Discriminant (FLD) functions were described in the previous Chapter. PCA is an unsupervised technique which returns a low-dimensional and uncorrelated sub-space that maximizes the variation across all classes of data in the input regardless of their relevance to the discrimination of the intended class structure. Accordingly, when the training dataset contains irrelevant variations of large magnitude, PCA will attempt to retain all of them, sometimes even at the expense of some of the useful information having relatively small variations.

In contrast to PCA, the FLD approach is a supervised technique that is capable of returning projections that extract some useful information. However, practical problems surface in using this technique when the amount of sample data is small compared to the dimensionality of the input space. Under these conditions, statistical techniques that are used to determine Fisher's projection matrix suffer from singularity issues due to the ill-conditioning of the within-cluster scatter matrix, $\mathbf{S}_w$ (2.10). The most practiced workaround for this problem is to first use PCA on all training data such that $\mathbf{S}_w$ in the projected space is non-singular [64]. Thereafter, the FLD method is used on the projected low-dimensional PCA space (2.11) using class label information to remove some of the unrelated variations.

However, it must be noted that this approach restricts the amount of relevant information to the extent captured by the first PCA projection.

3.6 Summary

In this chapter, theories, techniques and algorithms related to design and implementation of RBF networks were discussed. The general structure of a RBF network consists of two layers of processing nodes that play completely different roles in the network. The first layer of processing nodes uses non-linear basis functions to provide a mapping from input space onto an intermediate basis space spanned by the response of each basis function. From a classification point of view, these basis functions can be viewed as partitioning the input space into several non-overlapping regions, which in turn are mapped onto separate axes of the basis space. In the second layer, processing nodes map the basis space linearly onto the output. Therefore as a pattern classifier, performance of a RBF network depends largely on the ability of the first layer to represent the problem domain within the input space.

RBF networks have a number of beneficial features for pattern classification compared to other types of neural networks. Compared to MLP, RBF networks require relatively small number of learnable parameters to define the network. Other beneficial features include the availability of faster training algorithms that use statistical information of the input space and the close relationship with Bayes decision theory. However, as a problem's dimensionality increases, some of these advantages become practically unrealizable due to the limitations of techniques that are used to create and train the network. Several workarounds that can be divided into local and global approaches have been proposed to solve these problems in practical situations. The first attempts to find a solution by improving the representation of basis functions within the high-dimensional input space whereas the second such as PCA and FLD attempts to project the input onto some low dimensional space in which problems due to dimensionality are of less significance. However, both these approaches have several

limitations and therefore must be selected based on the requirements and nature of the classification problem.

CHAPTER 4

The Proposed Methods: New RBF Network Classifiers for Holistic Facial Expression Recognition

Holistic approaches to image recognition in general consider each pixel in the image to be a feature in the input space mainly because of the difficulties faced in explicit extraction of feature parameters from raw images. Therefore these approaches often require classification systems with properties different from those used in parametric feature based approaches to image recognition. Although the RBF networks described in the previous chapter have many of these desired properties, their capabilities sometimes fall short of the requirements for applications, like recognition of facial expressions, due to some of their architectural limitations and properties of the problem-domain. In this chapter, novel approaches to these problems using architectural enhancements to traditional RBF networks are proposed.

4.1 Introduction: Properties of the Problem Domain

When used for holistic recognition, facial images showing expressions create a different paradigm in high-dimensional pattern recognition with Radial Basis Function (RBF) network-based classifiers. The problem-domain for the classification is characterized by some important properties that include (i) a high dimensional input space as large as the image size, (ii) presence of noise, (iii) large variations in the input that are not related to facial expressions, and (iv) importance of different sets of features for discriminating between different expression classes. Also in practice the number of image samples available for training is often much smaller than the dimensionality of the input space.

It has been reported that the minimum resolution required by a human operator to identify a face in an image is at least 32×32 pixels whereas for machine recognition, the limit is around 64×64 pixels [23]. Therefore even at these lower limits, the input space for holistic recognition will consist of more than 4000 features leading to even larger dimensionality than in many other types of high-dimensional problems where RBF networks are widely used. One of the first implications of this high-dimensional input space is the large number of parameters in the underlying RBF network that must be computed from the training data using some statistical algorithm. However, in most practical situations the amount of available training data is limited to a few hundred images, which itself is insufficient for most multivariate statistical techniques due to the dimensionality involved.

Another condition that requires special attention in classifier design is the presence of large amounts of non-discriminative information in the input. At the pixel level, a facial image contains a vast amount of detail such as skin-texture / tone, skin contours and structural details of the face that makes a person's identity different from others. Details like these do not play any role in the description of emotions but nevertheless, represent significant variations in the input space. As discussed in the previous chapter these irrelevant variations affect the partitioning of the input space by basis functions in the RBF network and lead to poor recognition accuracy and generalization by the classification system. Furthermore, because these variations are often embedded with some of the useful information, removing them while retaining the latter also becomes a difficult task for most feature extraction and dimensionality-reduction methods.

Another characteristic of the problem, which is closely related to the above, is the different roles played by various facial regions in displaying different facial expressions. Due to this a particular set of features that separate two classes of facial expressions may not be effective for the separation of a third class. For instance, pixels around the mouth region play an important role in separating Anger and Happy expressions but provide little information for

separating between Anger and Sad expressions, which are mostly described by upper facial regions. However, even though the mouth region plays a minor role in separation of the latter, some significant variations may still be present in this region due to facial differences among people in test images. Therefore, the underlying classifiers must be able to differentiate between the useful and irrelevant features in the local domain of the classification problem. For instance, the classifier must be insensitive to the mouth region for separating of Anger and Sad expressions but at the same time be sensitive to the same for separating Anger and Happy expressions.

In general the above properties of high-dimensional classification problems raise multiple concerns that must be addressed properly in order to design effective RBF network based classifiers:

1. Effects of large dimensionality and small sample size.
    - A large network trained with a small set of samples is likely to "memorize" the training input rather than learn a mapping that can generalize the output [112]. The small number of training samples is likely to represent only a subset of the complex mapping domain and therefore the network is likely to be over-trained and will generalize poorly with respect to input patterns that the network has not seen during training.

    - It has also been shown that the number of samples required to effectively estimate a multivariate density increases exponentially with dimensionality of the input space [113][114]. Consequently the unsupervised algorithms that are used to model data distributions in the input space, and subsequently obtain the parameters of the basis functions become less reliable with the limited number of training samples compared to larger dimensionality of input space.

- The sample covariance matrices become singular when $N < d + 1$, where N is the number of training samples and d is the dimensionality of the input space. This prevents the use of Gaussian basis functions with their full covariance matrices (3.47) in the RBF network. To handle this some researchers have proposed techniques like SVD [109] and pooled covariance matrices [110] to find numerically regularized solutions for the inverse. However these techniques can capture only a small subset of principal variations in the input depending on the number of training samples. Therefore the solutions obtained from such methods are likely to be sub-optimal since the information captured by them may not necessarily represent all features required by the problem domain.

2. Influence of non-relevant features in the input.
    - Presence of irrelevant features causes the respective class-conditional data distributions in input space to stretch along directions that are less discriminative with respect to the intended class structure of facial expressions. As illustrated in the previous chapter, such conditions are likely to result in class boundaries that are determined from the minor axes of the Gaussian data spreads, thereby leaving a portion of data outside the region represented by the basis function.

Several recent developments of RBF networks that address the above issues were discussed in the previous chapter. However, when RBF network classifiers are used for holistic recognition of facial expressions from static facial images, problems related to the influence of irrelevant features need greater attention. As discussed earlier such irrelevant features arise commonly in facial images mainly due to facial structure differences among people in the training images. Additionally, the absence of a common set of features that can discriminate between all six expressions also contributes to the problem by creating variations that are irrelevant in the respective local expression domains, making the classifier

insensitive to such variations by using a global approach a difficult task. Instead, what is required is a set of basis functions that can be adopted to variations that are important within their local domains.

4.2 Nomenclature

In the following sections, organization of the input data is referred to in different contexts: as data vectors grouped according to their assigned class labels or as data vectors grouped according to their natural spread in the input space. In order to avoid any ambiguity between these two aspects of data organization, the following nomenclature is used.

- The term "*class*" refers to a group of data vectors having the same class label according to the intended class structure of the problem domain. Data distribution of a class in general is considered to be multi-modal.
- The term "*cluster*" refers to a uni-modal spread of a group of data vectors according to their natural spread in the input space. Therefore, a cluster may contain data vectors having different class labels.
- The term "*homogeneous cluster*" is used to identify a group of data vectors having a single class label and distributed uni-modally in the input space. Thus a single class of data may consist of several homogeneous clusters.

4.2.1 A New Approach: Basis Functions with Differentially Weighted Radius

The general structure of a RBF network classifier can be expressed as a two-tiered mapping

$$\left\{\mathbf{X} \in \mathbb{R}^d\right\} \xrightarrow{\boldsymbol{\varphi}(\cdot)} \left\{\mathbf{B} \in \mathbb{R}^h\right\} \xrightarrow{\mathbf{W}} \left\{\mathbf{Y} \in \mathbb{R}^q\right\} \tag{4.1}$$

where $\boldsymbol{\varphi}(\cdot)$ is a non-linear mapping from the input space $\left\{\mathbf{X} \in \mathbb{R}^d\right\}$ onto basis space $\left\{\mathbf{B} \in \mathbb{R}^h\right\}$ and $\mathbf{W}$ is usually a subsequent linear mapping from the basis space to output space $\left\{\mathbf{Y} \in \mathbb{R}^q\right\}$. The post-basis mapping represents linear hyper-planes in the basis space and therefore is capable of only linear discrimination. For this reason, the overall

performance of RBF network depends mostly on the first, i.e., the basis mapping and its ability to cast a non-linear problem in a such way that it becomes linearly separable in the basis space. Therefore, ideally the basis mapping should be able to represent the complete intrinsic dimensionality of the pattern recognition problem with a minimum number of non-overlapping basis functions.

Improvements that have been proposed to achieve these properties in RBF networks can be divided into two categories: global and local approaches. Global approaches attempt to address the above issues indirectly by projecting input space onto a low dimensional sub-space, which retains most of the information related to the problem domain. By projecting onto a lower dimension these techniques allow computationally feasible basis functions that use full covariance matrices to represent the spread of input data. Local approaches, on the other hand, attempt to modify the parameters of basis functions within the high-dimensional space to suit local properties of the problem domain. A majority of the local approaches therefore constrain the parameters for the Gaussian basis functions to reflect some of the principal variations in the available training input data. However, these basis functions may be a sub-optimal fit to the true data distribution.

As pointed out earlier, when used for holistic facial expression recognition, the benefits of using RBF networks are limited by the properties of the holistic input space and the lack of a sufficient number of training samples. Therefore in the following sections, modified basis functions and new training algorithms are developed in consideration of properties of the problem domain. However in contrast to previous methods, the proposed approaches use a combination of statistical and algorithmic techniques to determine basis function parameters that provide improved representation of the input in basis space even with small training data sets.

4.2.2 Spherical Basis Functions and Problems with the Euclidean Radius

Hyper-spherical basis functions are the simplest form of basis functions that can be used in RBF networks operating in a high-dimensional input space. The basis functions are defined only with two parameters $\{\boldsymbol{\mu}, \sigma\}$ and are given as

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right) \tag{4.2}$$

where

$$\|\mathbf{x}-\boldsymbol{\mu}_j\|^2 = \sum_{i=1}^{d}(x_i-\mu_{ij})^2 \tag{4.3}$$

is the Euclidean distance between the d-dimensional input $\mathbf{x}$, and the centre $\boldsymbol{\mu}_j$ of the j^{th} basis function. Parameter σ_j^2 in (4.2) is a smoothing constant which determines the overall radius of the basis function. In effect, it determines the hyper-volume boundary of the region in input space which is represented by the basis function.

The foremost difference in the response of (4.2) and a general Gaussian basis function (3.45) is related to the fact that the former depends only on the Euclidean distance between input vector and a prototype vector at the center of the basis function, whereas the latter depends on the Mahalanobis distance between the two vectors. This implies that spherical basis functions give equal weight for all features in the distance calculation (4.3) whereas in general Gaussian basis functions, features are weighted non-equally by their corresponding variance and covariance coefficients. As a result the first treats all features in input space as equally important in evaluating class membership and is therefore seldom capable of separating the relevant features in input space from the irrelevant. On the contrary, general Gaussian basis functions scale the emphasis on different features based on their local variations and therefore are more capable of representing those properties in the local problem domain.

4.2.3 A Differentially Weighted Radius for Spherical Basis Functions

The above interpretation of spherical and general Gaussian basis function responses suggests that performance of the former, in terms its ability to represent the relevant local variations of the input space can be improved by using a weighted sum of distances in (4.2) instead of the simple Euclidean distance in (4.3). The weighted distance can be defined as

$$\left\|\mathbf{x}-\boldsymbol{\mu}_j\right\|^2 = \sum_{i=1}^{d} \Theta_i \left(x_i - \mu_{ij}\right)^2 \tag{4.4}$$

$$= \left(\mathbf{x}-\boldsymbol{\mu}_j\right)^T \boldsymbol{\Theta} \left(\mathbf{x}-\boldsymbol{\mu}_j\right) \tag{4.5}$$

where Θ_i is the weight associated with the i^{th} feature of the input space and

$$\boldsymbol{\Theta} = \begin{bmatrix} \Theta_1 & 0 & \cdots & 0 \\ 0 & \Theta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Theta_d \end{bmatrix} \tag{4.6}$$

is a diagonal matrix having Θ_i as diagonal elements. In comparison to (4.3), weights associated with each feature in (4.4) can be used to specify their contribution to the distance measure based on two criteria; the extent of their variability and their relative importance in being able to separate homogeneous clusters within the local region of the input space. However, it is worth pointing out that Θ_i are not feature weights but scaling factors acting on the *difference* between the input vector and a prototype vector of the basis function. The objective of these weights is to differentially scale the *distance measure*, thereby emphasizing differences that are important in separation of pattern classes while compressing other variations with little relevance to the discrimination. For this reason, in the following development the weights $\left\{\Theta_i \in \mathbb{R}\right\}_{i=1}^{d}$ are referred to as "Discriminative Indices".

Using this idea, a general form of the "Differentially Weighted Radius Radial Basis Function (DWRRBF)" can be defined as

$$\phi_j\left(\mathbf{x} \mid \boldsymbol{\mu}_j, \sigma_j, \boldsymbol{\Theta}_j\right) = \exp\left(-\frac{\sum_{i=1}^{d} \Theta_{ij}\left(x_i - \mu_{ij}\right)^2}{2\sigma_j^2}\right) \tag{4.7}$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^d, \sigma_j \in \mathbb{R}$ are as defined for (4.2) and $\boldsymbol{\Theta}_j \in \mathbb{R}^d$ are Discriminative Indices of the j^{th} basis function. It is desirable that features which are less useful to the discrimination in the local region are assigned Discriminative Indices of relatively lower values, thereby reducing the contribution of such features to the distance measure. This makes the basis function (4.7) to be less sensitive to variations of irrelevant features compared to useful ones. Geometrically, this can be viewed as stretching a spherical basis function along feature axes which have larger but irrelevant variations to cover the wider spread of data along these axes.

It must be noted that if Discriminative Indices are associated only with the variance of each feature in the input space, the effect of (4.7) will be similar that of a Gaussian basis function with a diagonal covariance matrix containing the variance of each feature [81]. However, this approach will not be able to differentiate between variables that are useful for the discrimination from those that are not. Instead a more useful method to determine Discriminative Indices can be derived by extending the concept of Rayleigh coefficients [21]. The Rayleigh coefficient attempts to capture a subset of useful features from a noisy feature set by maximizing the useful energy of these features compared to the overall energy of feature space. Extending this concept, two criteria for computation of Discriminative Indices can be derived as follows:

1. **Variance based criterion**: Distribution of images within the holistic face image space is affected mainly by three types of features: (i) those describing facial expressions, (ii) those describing the subject's identity and (iii) those

describing the differences in the environment such as different levels of illumination, presence of shadows etc. To separate facial expression classes, the first type of features are more useful and should have a higher variance within all facial images of a single subject showing different facial expressions, compared to the other two types of features. On the other hand when a subset of facial images of different subjects having the same facial expression is considered, variance of features that describe the facial expression should be lower compared to features that describe the subject's identity and those related to the environmental conditions. Based on these properties the Discriminative Index Θ_i for the i^{th} feature can be computed using the variance criterion as

$$\Theta_i = \frac{\mathrm{var}(x_i) + \frac{1}{S}\sum_{p=1}^{S} \mathrm{var}\left((x_i)_{\mathbf{x}\in S_p}\right)}{2\frac{1}{h}\sum_{l=1}^{h} \mathrm{var}\left((x_i)_{\mathbf{x}\in \tilde{C}_l}\right)} \tag{4.8}$$

where $\mathrm{var}(\cdot)$ is the variance operator, $\tilde{C}_l, l = 1,\cdots,h$ are subsets of images in h homogeneous clusters having images of a single expression class (i.e. clusters of facial images represented by each basis function in the network), $S_p, p = 1,\cdots,S$ are subsets of images belonging to different subjects in the training image set, and $\mathrm{var}(x_i)$ is the variance of i^{th} feature in the training image set. The overall variance of the i^{th} feature, $\mathrm{var}(x_i)$ in (4.8) represents the variations due to environment conditions while the other two components represent the variations across different facial expressions and across different subjects.

2. **Mean-based criterion**: A homogeneous cluster $\tilde{C}_j$ of facial images showing the same facial expression is likely to be best separated by a set of

features that are compact within these images and separated widely from the rest. Using this idea the Discriminative Index Θ_{ij} of the i^{th} feature of the j^{th} basis function can be computed as

$$\Theta_{ij} = \frac{\frac{1}{\left(\tilde{h}\right)} \sum_{\boldsymbol{\mu}_l \in \left\{\tilde{C}^n\right\}} \left(\mu_{ij} - \mu_{il}\right)^2}{\mathrm{var}\left(\left(x_i\right)_{\mathbf{x} \in \tilde{C}_j}\right)} \tag{4.9}$$

where $\mathrm{var}(\cdot)$ is the variance operator, μ_{ij} is the i^{th} feature of the prototype vector of a homogeneous cluster of images $\tilde{C}_j$ represented by the j^{th} basis function $\phi_j(\cdot)$ and $\left\{\tilde{C}^n\right\}$ is the set of $\tilde{h}$, $\left(\tilde{h} < h\right)$ nearest basis function centers of $\phi_j(\cdot)$.

It must be noted that the *variance-based criterion* (4.8) computes a single set of Discriminative Indices, common to all basis functions whereas the *means-based criterion* (4.9) returns different sets of Discriminative Indices for each basis function in the network. Furthermore, Discriminative Indices returned by both methods can range from very small values to extremely large values and therefore if not normalized could lead to potential problems in computing the exponential function (4.7) due to rounding off errors. This problem can however be avoided by normalizing them to a vector of unit length as

$$\tilde{\Theta}_i = \frac{\Theta_i}{\sqrt{\boldsymbol{\Theta}^T \boldsymbol{\Theta}}} \tag{4.10}$$

where $\tilde{\Theta}_i$ is the normalized Discriminative Index of the i^{th} feature.

4.3 Creating and Training RBF Networks Using DWRRBF

Using the general form of DWRRBF in (4.7) the two tiered mapping of the RBF networks can be expressed in the matrix form (3.9) as

$$\mathbf{y} = \mathbf{W}\boldsymbol{\varphi}(\mathbf{x}) \tag{4.11}$$

and

$$\boldsymbol{\varphi}(\mathbf{x}) \equiv \left[\phi_0(\mathbf{x}),\ \phi_1(\mathbf{x} \mid \{\boldsymbol{\mu}_1, \sigma_1, \boldsymbol{\Theta}_1\}),\ \cdots\cdots,\ \phi_h(\mathbf{x} \mid \{\boldsymbol{\mu}_h, \sigma_h, \boldsymbol{\Theta}_h\})\right]^T \tag{4.12}$$

where $\phi_0(\mathbf{x}) = 1$ corresponds to the bias term of post-basis mapping, h is the number of basis functions and $\boldsymbol{\mu}_j$, σ_j, $\boldsymbol{\Theta}_j$ are parameters that define the j^{th} basis function. Each basis function in (4.12) can have its own vector of Discriminative Indices. The gradient-descent learning algorithm described in Section 4.3.2 allows learning these according to the local properties of each basis function. When the Discriminative Indices are computed using the variance-based method (4.8) all the basis functions in (4.12) are initialized to the same Discriminative Index vector returned by (4.8). On the other hand, if the means-based method (4.9) is used, each of the h basis functions in (4.12) is initialized to a different Discriminative Index vector.

The output layer of the RBF network defined in (4.11) consists of q output nodes, one for each of the q expression classes recognized by the network. In algorithms proposed here, a linear discriminant function represented by a $q \times (h+1)$ dimensional weight matrix $\mathbf{W}$ is used as the post-basis mapping. For a given input vector, the expression-class is defined by the output node which produces the largest response.

As described in the previous chapter, an RBF network-based classifier is commonly designed and trained in two stages. During the first stage the number of basis functions and their parameters are determined using some statistical procedure, such as the k-means clustering in the input space. The basis function parameters are then specified directly from the clusters obtained. Thereafter, the basis functions are kept fixed and, in the second stage, post-basis mapping is computed using a supervised procedure that accounts for the class

labels of the training input. In most instances, the least squares solution in (3.4) is used to generate a single step solution for this purpose.

In contrast to the above procedure, an RBF network consisting of the proposed DWRRBF requires a more integrated approach for the creation and training of the network. With the introduction of Discriminative Indices, clustering procedures must take them into account to compute the new weighted-distance criterion. On the other hand, the Discriminative Indices themselves, and therefore the weighted distance criterion, depend on the outcome of the clustering algorithm. Therefore, in the proposed training algorithms both these tasks are carried out using an iterative optimization procedure, accounting for the inter-dependency described above. To achieve this integration, the proposed algorithm starts with the minimum number of basis functions, i.e. one per each expression class. Thereafter, the iterative procedure increments the number of basis functions until the required performance goals are met. The major steps involved in this iterative procedure are presented in the following section.

4.3.1 The Integrated Training Algorithm

We assume that a training image data set is given consisting of N pairs of labeled samples $\{\mathbf{P}\} \equiv \{\mathbf{x}_j, \mathbf{t}_j\}_{j=1}^{N}$, where $\mathbf{x}_j$ is a $d \times 1$ dimensional vectorized image showing a single facial expression and $\mathbf{t}_j$ is a corresponding q dimensional column vector consisting of the class label of $\mathbf{x}_j$ are available. The elements in target vector $\mathbf{t}_j$ are encoded as follows.

$$t_{ij} = \begin{cases} 1 & \text{if } i = \text{class label of } \mathbf{x}_j \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

All source images which are used to create these training samples are assumed to be normalized with respect to lighting conditions, scaling, rotation and registration. The steps in the training algorithm are:

1. Divide the training data into h $(h = q)$, homogeneous subsets $\{\tilde{C}_j\}_{j=1}^{h}$ according to their given expression class labels. These subsets form the initial set of homogeneous data clusters from which the initial estimates for Discriminative Indices and basis function parameters are computed.

   Let $\mathrm{Cnt}(c_k, \{I\})$ be a function that returns the number of images with class label c_k from a set of facial images $\{I\}$. Also let h_{c_k}, $c_k = 1, \cdots, q$ be the number of homogeneous data clusters with images having the class label c_k according to the current configuration of the network. Since initially there is only one cluster per each expression class, initialize $h_{c_k} = 1$ for $c_k = 1, \cdots, q$.

   Let *min_Csize* be a generalization parameter that specifies the maximum allowed number of misclassified training images per expression class at convergence of the training procedure.

2. Using h homogeneous clusters, create a RBF network with h basis functions. First compute the corresponding normalized Discriminative Indices vectors $\{\boldsymbol{\Theta}_j\}_{j=1}^{h}$ for the basis functions. Note that the mean-based criterion (4.9) returns a different Discriminative Indices vector to initialize each of the h basis functions. If the variance-based criteria (4.8) is used instead, then initialize all h Discriminative Indices vectors to the vector returned by (4.8). Next compute the parameters $\{\boldsymbol{\mu}_j, \sigma_j\}_{j=1}^{h}$ for basis functions according to the current set of homogeneous data clusters as

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{\mathbf{x}_l \in \tilde{C}_j} \mathbf{x}_l \tag{4.14}$$

and

$$\sigma_j^2 = \frac{1}{2(h-1)} \sum_{\substack{l=1, \\ l \neq j}}^{h} (\boldsymbol{\mu}_l - \boldsymbol{\mu}_j)^T (\boldsymbol{\mu}_l - \boldsymbol{\mu}_j) \tag{4.15}$$

where N_j is the number of training inputs associated with the homogeneous cluster $\tilde{C}_j$. Parameter σ_j^2, controls the overall radius of the basis function and is determined heuristically as half of the average distance between cluster j and the rest. Then compute the weight matrix $\mathbf{W}$ for the post-basis mapping using (3.41).

3. Use the iterative learning process, which is described later (in Section 4.3.2), to train all learnable parameters based on the current set of basis functions and Discriminative Indices. All learnable parameters in the network, i.e. those in the basis functions as well as those in the post-basis mapping, are trained at the same time.

4. Check whether the stopping criterion, described in Section 4.3.3 has been reached or the maximum number of epochs (c_epoch) has been reached. Otherwise continue the gradient descent learning process as in step 3.

5. Present all training patterns, $\{\mathbf{x}_j\}_{j=1}^N$ to the network and compute their corresponding output vectors $\{\mathbf{y}_j\}_{j=1}^N$. Then convert each output vector $\mathbf{y}_j$ to encode its output class labels as

$$y_{ij} = \begin{cases} 1 & \text{if } y_{ij} \geq y_{lj} \text{ for } l = 1, \cdots, k;\ l \neq i \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

where y_{ij} is the i^{th} element of the vector $\mathbf{y}_j$.

6. Compare $\{\mathbf{y}_i\}_{i=1}^{N}$ with their corresponding target values $\{\mathbf{t}_i\}_{i=1}^{N}$, and split the training data set $\{\mathbf{P}\}$ into two subsets $\{\mathbf{P}^C\}$ and $\{\mathbf{P}^M\}$, respectively containing correctly classified and misclassified training samples.

7. If $\mathrm{Cnt}\left(c_k, \{\mathbf{P}^M\}\right) \leq min\_Csize$ (see Section 4.3.4) for all $c_k = 1, \cdots, q$ or $\{\mathbf{P}^M\} = \{\mathbf{P}^M\}^{previous}$ where $\{\mathbf{P}^M\}^{previous}$ is the misclassified data set in the previous iteration, then stop.

8. In the following steps new basis functions are added to the network based on expression-classes for which $\mathrm{Cnt}\left(c_k, \{\mathbf{P}^M\}\right) > min\_Csize$. First using associated class labels, partition $\{\mathbf{P}^M\}$ into q' homogeneous clusters with each cluster containing only misclassified images of a single expression class. Note that $q' < q$ if, for some expression classes, the number of misclassified images in the training set is less than min_Csize.

9. For each expression class c_k for which $\mathrm{Cnt}\left(c_k, \{\mathbf{P}^M\}\right) > min\_Csize$, increment the number of basis functions in the network by re-partitioning the set of training images with class label c_k into $h_{c_k} + 1$ number of homogeneous data clusters. Use the splitting criteria described later in Section 4.3.4 to partition input data using parameters of the existing h_{c_k} basis functions and the misclassified data in $\{\mathbf{P}^M\}$. A total of q' basis functions will be added to the network.

10. Update the total number of basis functions (h) in the network

$$h = h^{current} + q' \tag{4.17}$$

and then return to step 2

Two parameters c_epoch and min_Csize are used to control the running of the training process. The first sets the upper bound for the maximum number of epochs in the gradient descent learning of the basis function parameters, that is especially useful at slow update rates. On the other hand the second parameter min_Csize controls the desired accuracy of the network by defining the threshold for the number of misclassified training samples in each expression class that would require re-partitioning of data clusters of the expression class. During steps 8 and 9 of the training algorithm, it prevents new basis functions from being added based on potential outliers in the training dataset.

4.3.2 Iterative Learning of Network Parameters

During step 3 of the integrated learning algorithm described in the previous section, all parameters that define the RBF network, except the basis function centers, are further fine-tuned using an iterative learning procedure. Two parameters of basis functions, Discriminative Indices $\{\boldsymbol{\Theta}\}$ and smoothing parameters $\{\sigma\}$, are learned using a gradient descent approach while the post-basis mapping $\mathbf{W}$ is updated using the least square solution on each iteration of the learning process. Using (3.33), (3.37) and the modified basis function (4.7), the batch update rule for the smoothing parameter σ_j of the j^{th} basis function can be expressed as

$$\sigma_j^{\alpha+1} = \sigma_j^{\alpha} - \eta_1 \frac{\partial E(\mathbf{X})}{\partial \sigma_j} \tag{4.18}$$

and

$$\frac{\partial E(\mathbf{X})}{\partial \sigma_j} = \frac{1}{N}\sum_{p=1}^{N}\sum_{l=1}^{q}\left(y_{lp} - t_{lp}\right) w_{lj} \exp\left(-\frac{\sum_{i=1}^{d}\Theta_{ij}\left(x_{ip} - \mu_{ij}\right)^2}{2\sigma_j^2}\right)\frac{\sum_{r=1}^{d}\Theta_{rj}\left(x_{rp} - \mu_{rj}\right)^2}{\sigma_j^3} \tag{4.19}$$

where α is the epoch number, η_1 is the learning rate, Θ_{ij} is the i^{th} element of the Discriminative Indices vector associated with the j^{th} basis function, y_{lp}, t_{lp} respectively are the l^{th} elements of the output vector $\mathbf{y}_p$ and the target vector $\mathbf{t}_p$ corresponding to input vector $\mathbf{x}_p$ and N is the number of samples in the training data set.

Similarly the update rule for the i^{th} element Θ_{ij} of the Discriminative Indices vector $\mathbf{\Theta}_j$ associated with the j^{th} basis function $\phi_j(\cdot)$ can be expressed as

$$\Theta_{ij}^{\alpha+1} = \Theta_{ij}^{\alpha} - \eta_2 \frac{\partial E(\mathbf{X})}{\partial \Theta_{ij}} \tag{4.20}$$

and

$$\frac{\partial E(\mathbf{X})}{\partial \Theta_{ij}} = -\frac{1}{N}\sum_{p=1}^{N}\sum_{l=1}^{q}\left(y_{lp} - t_{lp}\right) w_{lj} \exp\left(-\frac{\sum_{r=1}^{d}\Theta_{rj}\left(x_{rp} - \mu_{rj}\right)^2}{2\sigma_j^2}\right)\frac{\left(x_{ip} - \mu_{ij}\right)^2}{2\sigma_j^2} \tag{4.21}$$

where η_2 is the learning rate and rest of the parameters are as defined for (4.19).

The post basis mapping $\mathbf{W}$ is updated using the least square solution (3.41). Thus after updating the parameters of the basis functions, the post-basis mapping for the next iteration can be computed as

$$\mathbf{W}^{\alpha+1} = \left(\mathbf{\Phi}^{\alpha+1}\right)^{\dagger}\mathbf{T} \tag{4.22}$$

where $\left(\mathbf{\Phi}^{\alpha+1}\right)^{\dagger}$ is the pseudo inverse of the $h \times N$ matrix

$$\Phi^{\alpha+1} = \left[\boldsymbol{\varphi}^{\alpha+1}\left(\mathbf{x}_1\right), \boldsymbol{\varphi}^{\alpha+1}\left(\mathbf{x}_2\right), \cdots, \boldsymbol{\varphi}^{\alpha+1}\left(\mathbf{x}_N\right) \right] \tag{4.23}$$

containing responses of basis functions to input vectors $\left\{\mathbf{x}_p\right\}_{p=1}^{N}$, and

$$\mathbf{T} = \left[\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N\right] \tag{4.24}$$

is a q x N matrix containing their corresponding target vectors. It must be noted that basis function outputs in (4.23) are computed after updating their parameters using (4.18) and (4.20).

It is important to note that besides using the weighted distance in the basis functions, the smoothing parameter σ_j^2 must still be present in the basis functions as it plays an important role in the partitioning of input space by the respective basis functions. This parameter determines the overall radius of the response region of a basis function and therefore controls the extent of overlap among regions represented by different basis functions. A larger radius will include more variations of features within a basis function but only at the risk of overlapping with another representing a different class of data. A smaller value on the other hand will cover only a small segment of input variations and therefore will cause a basis function to represent only a subset of images in a local region. Since the initial estimate for smoothing parameters computed in (4.15) pays little attention to these considerations, these parameters are further trained according to the local properties of the respective basis functions.

4.3.3 Stopping Criteria for Gradient Descent Learning

The error back propagation algorithm used to train the network parameters is subject to the well known risks of overtraining and getting trapped in local minima of the error surface. The risk of overtraining is even higher in the proposed paradigm due to the limited number of training samples compared to the number of parameters to be learned. Furthermore, when there are insufficient numbers of basis functions to represent the complete organization of

training data, the above gradient-descent algorithm may cause some basis functions to evolve with larger radii to cover regions that could otherwise be more generally represented by using additional basis functions with smaller radii [81]. Although large radii could tend to reduce the overall training error, generalization properties of the network will be affected due to the improperly represented input space. Basis functions with large radii are likely to occur more during the initial cycles of the algorithm outlined in Section 4.3.1. When the number of basis functions is insufficient to represent the entire spread of the multimodal distribution in input, the gradient-descent procedure will attempt to reduce the overall training error by enlarging the region covered by some of the basis functions. Therefore in order to avoid basis functions with larger radii, the stopping criteria for gradient decent learning process should be able to terminate prematurely allowing the creation of additional basis functions for better representation of the input.

To address the above concerns the proposed algorithm uses a two-tiered approach, that includes an upper bound for the number of epochs in the gradient decent learning algorithm and the use of a k-leave out cross-validation method [115]. The k-leave out cross-validation reduces the risk of the network not being able to generalize by checking the performance against an independent set of validation data that is not included in the training set. The validation data set typically consists of about 5% of the images selected randomly from the training image set. The network is trained only using the balance 95% of the input data and at the end of each epoch, the training error is computed for both data sets. The training stops when the error with respect to validation set starts to increase while having a negative gradient with respect to the training set.

4.3.4 Splitting Criterion for Addition of New Basis Functions

During the early stages of the integrated training procedure described in Section 4.3.1, where the number of basis functions in the network is unlikely to be sufficient in order to represent

the input space completely as homogeneous clusters, Step 6 of the algorithm will be reached with a significant number of training data being misclassified by the network. Here the network must be provided with additional basis functions to represent input regions containing misclassified training data. However simply creating new basis functions based only on clusters in the misclassified set $\left\{\mathbf{P}^M\right\}$ may not lead to the best results for two reasons. First there is no guarantee that all data points of the misclassified set are from a single compact region of the input space and therefore representing all of them using a single homogeneous basis function could be difficult. Second, the addition of new basis functions is likely to affect the overall cluster membership within the input space, and therefore parameters of other basis functions too need to be re-adjusted. Hence the misclassified samples are used only to compute initial estimates for homogeneous data clusters associated with the new basis functions. A version of the k-means algorithm, modified to incorporate the weighted distance is then used to re-partition the input space to re-determine the cluster membership of all training inputs.

The splitting criterion for creation of new basis functions starts by making an initial estimate of the homogeneous cluster centers for new basis functions from the misclassified inputs in $\left\{\mathbf{P}^M\right\}$. For each class c_k where $\mathrm{Cnt}\left(c_k, \left\{\mathbf{P}^M\right\}\right) > min\_Csize$ an initial estimate for a new cluster center is selected as the input data vector $\mathbf{x}_{c_k} \in \tilde{C}_{c_k}$ having

$$\sum_{\mathbf{x}\in\tilde{C}_{c_k}} \left\|\mathbf{x}-\mathbf{x}_{c_k}\right\|^2 \leq \sum_{\mathbf{x}_p\in\tilde{C}_{c_k}} \left\|\mathbf{x}-\mathbf{x}_p\right\|^2 \tag{4.25}$$

for all $p \neq c_k$, $\mathbf{x}, \mathbf{x}_p \in \tilde{C}_{c_k}$, and $\tilde{C}_{c_k} \in \mathbf{P}^M$ representing misclassified inputs of class c_k. The criterion in (4.25) selects the misclassified input with the shortest distance to all other misclassified inputs of the same class and thereby reduces the risk of selecting a potential outlier as the candidate. The newly selected cluster centers and the existing basis function

centers representing data of their respective classes are used as the initial cluster centers of the modified k-means clustering algorithm.

Since DWRRBFs associate basis functions to the input using a weighted distance, the same weighted distance criteria must be used in the k-means clustering algorithm that is used to determine data clusters represented by those basis functions. However, such weights (i.e. Discriminative Indices) themselves would depend on the outcome of the clustering algorithm and therefore an initial estimate must be made at this point, based on the Discriminative Indices associated with the existing basis functions. Therefore, in order to re-partition input data belonging to expression class c_k, an estimate of Discriminative Indices vector is computed as

$$\hat{\boldsymbol{\Theta}}_{c_k} = \frac{1}{h_{c_k}} \sum_{j=1}^{h_{c_k}} \boldsymbol{\Theta}_j^{c_k} \tag{4.26}$$

where h_{c_k} is the current number of basis functions representing data belonging to class c_k and $\left\{ \boldsymbol{\Theta}_1^{c_k}, \boldsymbol{\Theta}_2^{c_k}, \cdots, \boldsymbol{\Theta}_{h_{ck}}^{c_k} \right\}$ are the set of Discriminative Indices vectors associated with those basis functions.

Finally a modified version of the standard k-means algorithm described in the previous chapter (Section 3.4.3) is invoked to iteratively determine the new partitioning. The clustering criterion function (3.25) is modified to use the weighted distance as

$$J(S) = \sum_{j=1}^{h_{c_k}+1} \sum_{\mathbf{x} \in \tilde{C}_j^{c_k}} \sum_{i=1}^{d} \hat{\Theta}_i^{c_k} \left(x_i - \mu_{ij} \right)^2 \tag{4.27}$$

where h_{c_k} is the current number of basis functions representing data of class c_k in the network, $\tilde{C}_j^{c_k}$ is the j^{th} homogenous cluster with data of class c_k and with $\boldsymbol{\mu}_j$ as the cluster center and $\hat{\boldsymbol{\Theta}}^{c_k}$ is the estimated Discriminative Indices vector computed in (4.26).

4.4 Addressing the Problem of Locally Important Features

We define locally important features as those that are important only for the discrimination of some of the facial expression classes. One characteristic that makes facial expression recognition different from other classification problems is the different roles played by facial regions in displaying facial expressions. Consequently, a facial region that is important in the display of one class of facial expressions may be of little significance in the display of another. In the input feature space, this leads to features that are important locally, i.e. only for the separation of one particular expression class from another but not for the rest of the expression classes in the problem domain. For example, features around the mouth region are significant in separation between Sad and Happy expressions but play a relatively minor role in separation between Sad and Angry expressions (Figure 4.1).

(a) (b) (c)

Figure 4.1: Different roles played by the mouth region during (a) Sad, (b) Happy and (c) Angry expressions. Note that there is significant difference in the mouth region between Sad and Happy expression compared to the differences between Sad and Angry expressions.

Discriminative Indices computed in (4.8) and (4.9) may fail to emphasize the locally important features due to the averaging effect. Since the Discriminative Indices are computed with respect to all expression classes, feature variations that are significant only for a few classes of facial expressions will be averaged by their corresponding smaller variation in the rest of the expression classes. Therefore the Discriminative Indices, and hence the weighted distance criterion of the basis functions will still be dominated by the large variations that are present in the majority of the expression classes. Additionally there is no guarantee that the gradient-descent optimization of basis function parameters will

converge to optimal values within the allowed maximum number of epochs. This is because updates to the Discriminative Indices representing these may be slow due to the averaging effect mentioned above.

4.4.1 A Hierarchical Classification System

One approach to address the above issue of locally important features is to arrange the classification problem hierarchically so that at each level the classification is made according to a common subset of features belonging to some prominent facial regions. An approach similar to this was first used on holistic face images by Daw-Tung et. al.[58]. The classification was done in two levels with the first level having four combined expression categories ({Happy, Disgust}, {Anger, Surprise, Fear}, {Neutral}, {Sadness}) based on the mouth region. In the second level, these combined categories were further subdivided into their respective expression classes based on features of the eye region. The hierarchy itself was decided based on visual appearance of the shape of the mouth and eyes in the description of these expression classes. The authors were able to record a near-perfect accuracy for the first level using separate image segments of the mouth region. However, using image segments from the eye-region the classifier at the second level did not perform well, yielding only an even chance in recognition.

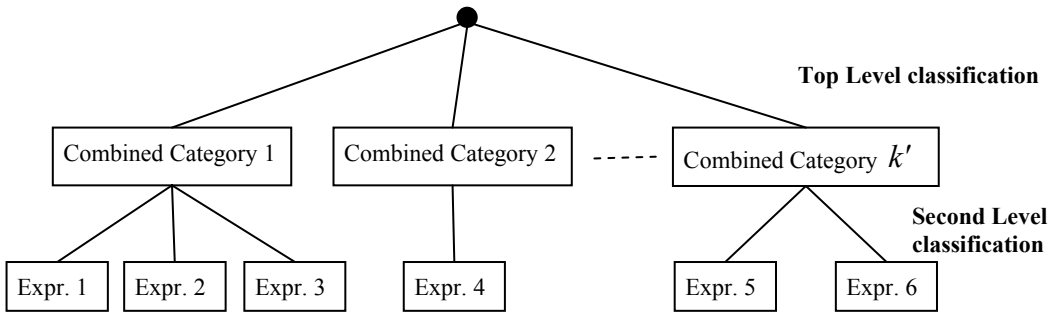


Figure 4.2 : An example of hierarchical classification. At the top level the input is classified into one of k' combined categories of expressions. At the second level, combined categories are further discriminated into individual expression classes.

In contrast to Daw-Tung's approach, for DWRRBF networks it is proposed to experimentally determine the hierarchy using a representative sample of facial images taken

from the training data set. Instead of using the visual appearance, combined expression classes in the top level of the two level hierarchy (Figure 4.2) are determined by the merger of expression classes having higher percentage of confusion in a non-hierarchical classifier. Therefore, compared to Daw-Tung's method, the proposed approach allows the network itself to determine the classification hierarchy based on its ability to use the features that are important in the local separation of expression classes.

4.5 DWRRBF with Multiple Function Boundaries

A property that is likely to affect the performance of DWRRBF networks is the different extent of separation between basis functions in the input space. In general, different facial regions that describe facial expression have different degrees of variability. For instance, features in the mouth region of the face are highly variable compared to features that describe the eye region. Therefore two basis functions, representing expressions that are separated predominantly by the mouth region (e.g. Sad and Happy expressions) will have a wider separation between them compared to two basis functions representing expressions that are discriminated predominantly by the eye region (e.g. Sad and Angry expressions).

The Discriminative Indices used in DWRRBFs are expected to account for the above properties by differentially scaling the distances measured within basis functions. However, in practice this solution may not address the problem completely due to the following reasons. First, owing to reasons described in Section 4.4 the Discriminative Indices may not be able to completely emphasize all features that are important (especially those important only for a few expressions) for the separation of a basis function from its neighbors. Second, the Discriminative Indices scale only along directions of the feature axes whereas major axes of the data distribution are unlikely to be oriented parallel to these feature axes. Finally, when basis functions are separated by different extents, the boundary of the j^{th} basis function determined by σ_j would tend to converge based on the weighted distance to its

neighbor with the narrowest separation. Therefore, in spite of the differential scaling by Discriminative Indices, there is no guarantee that the boundary of the basis function would completely enclose the region in input space which the basis function is expected to represent.

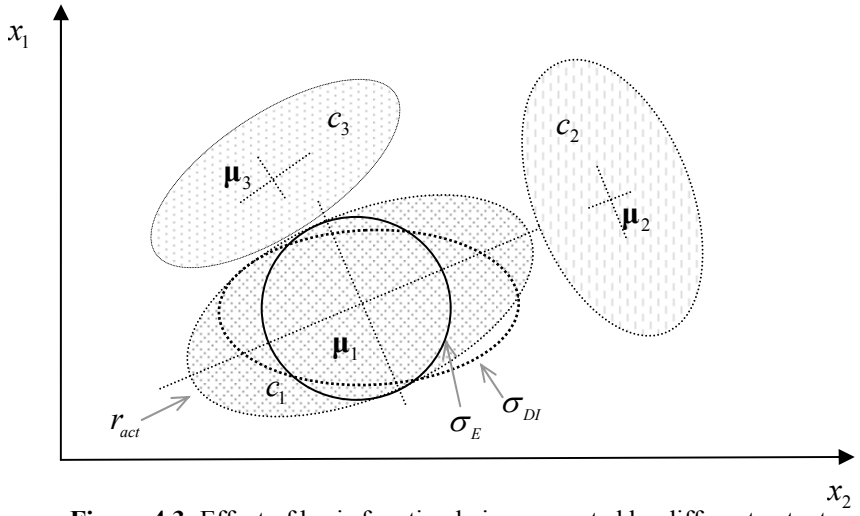


Figure 4.3: Effect of basis function being separated by different extents.

The illustration in Figure 4.3 demonstrates the above phenomenon using three basis functions in two-dimensional input space. For simplicity, each basis function is assumed to represent a homogeneous cluster of data from a single class, having a uni-modal Gaussian distribution. Suppose for class c_1 the true data distribution in input space is marked by the boundary r_{act}. Note that data in c_1 has a wider spread along the feature axis x_2 as compared to x_1. Now, if a simple spherical basis function is used to represent this homogeneous cluster, the radius will be determined as σ_E so as to not overlap with its nearest neighbour c_3 regardless of the wider spread towards c_2. A DWRRBF on the other hand will attempt to scale count for the differences in variability of x_1 and x_2 by assigning a lower weight to the distance measured along x_2 compared to that assigned along x_1. Graphically, this can be viewed as stretching the basis function along x_2 while compressing along x_1, thereby leading to a boundary σ_{DI} that encloses a larger portion of the data compared to the spherical basis function. Since Discriminative Indices in DWRRBF act only along the

directions of feature axes, the narrow separation between c_1, c_3 and the differences in orientations of r_{act} and σ_{DI}, causes the basis function boundary to be limited, and it cannot enclose the complete spread of c_1 in input space.

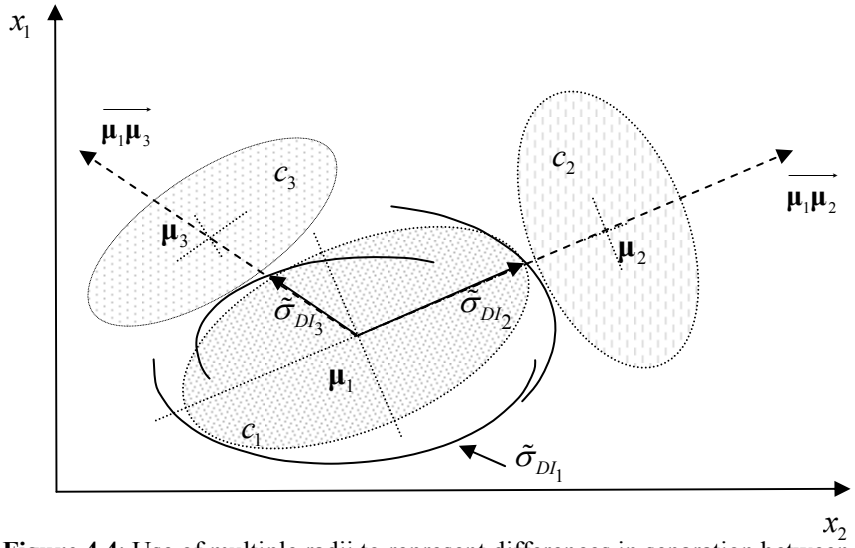


Figure 4.4: Use of multiple radii to represent differences in separation between basis functions.

The above problem can be solved if different boundaries are used for the separation between c_1, c_3 and c_1, c_2 (Figure 4.4). The basis function boundary is represented using a set of boundary segments with different radii according to the extent of data spread and the separation between homogeneous data clusters. For instance, the narrow separation between c_1, c_3 (Figure 4.4) could be maintained using the shorter radius $\tilde{\sigma}_{DI_3}$ while using a larger radius $\tilde{\sigma}_{DI_2}$ to enclose the wider spread of data towards the separation of c_1, c_2. Then, given an input vector, the most appropriate basis function radius can be selected based on the relative position of the input vector with respect to the basis functions. For instance, if the input vector is oriented towards $\overrightarrow{\boldsymbol{\mu}_1\boldsymbol{\mu}_2}$ parameter $\tilde{\sigma}_{DI_2}$ can be used to define the basis function boundary while, for any input oriented towards $\overrightarrow{\boldsymbol{\mu}_1\boldsymbol{\mu}_3}$, the radius $\tilde{\sigma}_{DI_3}$ can be selected instead. Note that, in addition to $\tilde{\sigma}_{DI_2}$ and $\tilde{\sigma}_{DI_3}$ that defines the separation for c_2

and c_3 respectively, another boundary segment $\tilde{\sigma}_{DI_1}$ is also required to act as the *default* boundary of the basis function. The default boundary is selected when the input vector is not oriented towards any of the surrounding basis functions.

Extending the concept shown in Figure 4.4 to $k-$nearest basis functions surrounding a homogeneous cluster c_1, the function boundary will be modeled by a set of boundary segments with $k+1$ different radii. In high-dimensional space, each of these radii will represent a segment of a hyper-elliptic boundary. The introduction of multiple radii adds another level of non-linearity to the basis function, so as to further partition the region represented by the basis function into different sub-regions. Each such intra-basis partition can be trained according to local properties of the problem domain, making the basis function more versatile in discriminating different classes of expressions with different local properties.

4.5.1 A New Nomenclature

It is noteworthy that with the introduction of multiple radii, one may no longer use the term "radial basis function". Instead, due to cloud like shapes formed by the set of boundary segments in the basis function, we will refer to them as "Cloud Basis Functions (CBF). Accordingly each boundary segment in the CBF is referred to as a "Cloud Segment" (CS), and the radius of the boundary segment is referred to as a "Cloud Segment Radius" (CSR).

4.6 Cloud Basis Function Networks

In accordance with the above description, a node in a Cloud Basis Function network can be represented by modifying the DWRRBF in (4.7) to include a selection criterion function for the most appropriate radius as

$$\phi_j\left(\mathbf{x}\,|\,\boldsymbol{\mu}_j,\{\tilde{\sigma}\}_j,\boldsymbol{\Theta}_j\right)=\exp\left(-\frac{\sum_{i=1}^{d}\Theta_{ij}\left(x_i-\mu_{ij}\right)^2}{2\left(\mathrm{Sel}\left(\{\tilde{\sigma}\}_j\,|\,\mathbf{x}\right)\right)^2}\right) \tag{4.28}$$

where $\boldsymbol{\mu}_j$, $\boldsymbol{\Theta}_j$ respectively are the center and Discriminative Indices vector associated with the j^{th} basis function $\phi_j(\cdot)$ and $\mathrm{Sel}\left(\{\tilde{\sigma}\}_j\,|\,\mathbf{x}\right)$ is the Radius Selection criterion Function (RSF) that, for a given input vector $\mathbf{x}$, returns the most appropriate radius from a set of radii $\{\tilde{\sigma}\}_j$ associated with the basis function.

4.6.1 Selection of the Most Appropriate Radius

Each value of CSR from the set $\{\tilde{\sigma}\}_j$ associated with the j^{th} CBF, represents part of a hyper-elliptic boundary that separates the region represented by the basis function towards the direction of one of its neighbors (Figure 4.4). An algorithmically efficient way of storing this association is to relate each CSR, $\tilde{\sigma}_k$ to a reference vector from the center $\boldsymbol{\mu}_j$ of the j^{th} basis function and the center $\boldsymbol{\mu}_k$ of its k^{th} neighbor as $\vec{l}_k=\overrightarrow{\boldsymbol{\mu}_j\boldsymbol{\mu}_k}$. This representation lets the most appropriate radius be selected according to the angle between a vector from the basis center to the input vector $\left(\vec{l}=\overrightarrow{\boldsymbol{\mu}_j\mathbf{x}}\right)$ and the respective reference vectors associated with each CSR. Therefore, using the inner-product of the two vectors to represent cosine of the angle between them, the Radius Selectivity criterion Function (RSF) in (4.28) can be expressed as

$$\mathrm{Sel}\left(\{\tilde{\sigma}\}_j\,|\,\mathbf{x}\right)=\tilde{\sigma}_k \tag{4.29}$$

where

$$k=\arg\min\left(\cos^{-1}\left(\frac{\left(\mathbf{x}-\boldsymbol{\mu}_j\right)^T\left(\boldsymbol{\mu}_p-\boldsymbol{\mu}_j\right)}{\left\|\mathbf{x}-\boldsymbol{\mu}_j\right\|\cdot\left\|\boldsymbol{\mu}_p-\boldsymbol{\mu}_j\right\|}\right)\right)\quad \mathrm{p}=1,2,\cdots,k' \tag{4.30}$$

and

$$\{\tilde{\sigma}\}_j \equiv \{\tilde{\sigma}_0, \tilde{\sigma}_1, \tilde{\sigma}_2, \cdots, \tilde{\sigma}_{k'}\} \tag{4.31}$$

is a set of $k'+1$ CSRs associated with the j^{th} basis function, $\boldsymbol{\mu}_j$ is the center of the j^{th} basis function and $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_{k'}\}$ are corresponding centers of neighboring basis functions. It must be noted that the default radius $\tilde{\sigma}_0 \in \{\tilde{\sigma}\}_j$ is selected when the input vector is not oriented towards any of the neighboring basis functions. This condition can be determined by having a pre-determined maximum threshold for the angle computed in (4.30). When the minimum angle between an input vector and any of the reference vectors exceeds this threshold, the default radius can be selected.

4.6.2 Selection of k'-Nearest Basis Functions

The selection of the k'-nearest basis function for (4.29) plays a crucial role in the overall performance of a CBF. For instance choosing many CSRs in the same direction with small angles between them would bring little benefit to a CBF compared to a smaller set of CSRs distributed evenly around the basis function. The former is likely to have similar $\tilde{\sigma}$ values for all CSRs, thereby resembling the behavior of the DWRRBF with a single boundary. Hence, reference vectors for CSRs must be chosen so that they are as widely separated in angle as possible.

For a given basis function $\phi_j(\cdot)$ a simple approach to determine the above is to start with all $h-1$ basis functions as potential candidates. Thereafter the k'-nearest and most widely separated basis functions among them may be determined by iteratively eliminating basis functions using the weighted distance to $\boldsymbol{\mu}_j$ and the angle with respect to the other clusters in the neighborhood. The procedure is outlined below and used in the experiments described later in Chapter 6 (Section 6.3.2).

1. Let $H \equiv \left\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{j-1}, \boldsymbol{\mu}_{j+1}, \cdots, \boldsymbol{\mu}_h\right\}$ be the centers of $h-1$ basis functions in the network, except the j^{th} basis function for which the k'-directions of CSRs are to be determined. Let $\text{Dist}\left(\boldsymbol{\mu}_l\right)$ be a function that computes the weighted distance from the center $\boldsymbol{\mu}_j$ of the j^{th} basis function to $\boldsymbol{\mu}_l$ as

$$\text{Dist}\left(\boldsymbol{\mu}_l\right) = \sum_{i=1}^{d} \Theta_{ij} \left(\mu_{il} - \mu_{ij}\right)^2 \tag{4.32}$$

where $\boldsymbol{\Theta}_j$ is the Discriminative Indices vector associated with the j^{th} basis function.

2. Let h' be the current number of basis centers in H. If $h' \leq k'$ (Section 6.3.2) then stop.

3. Find the two basis centers $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_t$ with smallest angles between them:

$$\cos^{-1}\left(\frac{\left(\boldsymbol{\mu}_p - \boldsymbol{\mu}_j\right)^T \left(\boldsymbol{\mu}_t - \boldsymbol{\mu}_j\right)}{\left\|\boldsymbol{\mu}_p - \boldsymbol{\mu}_j\right\| \cdot \left\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_j\right\|}\right) \leq \cos^{-1}\left(\frac{\left(\boldsymbol{\mu}_r - \boldsymbol{\mu}_j\right)^T \left(\boldsymbol{\mu}_s - \boldsymbol{\mu}_j\right)}{\left\|\boldsymbol{\mu}_r - \boldsymbol{\mu}_j\right\| \cdot \left\|\boldsymbol{\mu}_s - \boldsymbol{\mu}_j\right\|}\right) \tag{4.33}$$

for $\boldsymbol{\mu}_p, \boldsymbol{\mu}_t \in H$, $\boldsymbol{\mu}_r, \boldsymbol{\mu}_s \in H$ and $p \neq r, t \neq s$.

4. Two centers $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_t$ returned in step 3, represent the two basis centers having the smallest angle between them with respect to the center of $\phi_j(\cdot)$. One of the two can be eliminated based on the longest weighted distance from the center of the j^{th} basis function. Hence if

$$\text{Dist}\left(\boldsymbol{\mu}_p\right) > \textbf{Dist}\left(\boldsymbol{\mu}_t\right) \tag{4.34}$$

then $\boldsymbol{\mu}_p$ is eliminated from H and H is recomputed as

$$H = H^{current} - \boldsymbol{\mu}_p . \tag{4.35}$$

Otherwise $\boldsymbol{\mu}_t$ is eliminated from H and H is recomputed

$$H = H^{current} - \boldsymbol{\mu}_t \tag{4.36}$$

5. Go to step 2.

When choosing the k' nearest neighbors, the above algorithm uses the widest angle between neighboring basis functions as the primary criterion for retention as this would allow the selected basis functions to be distributed widely around the basis function for which the CSRs are determined. When there are multiple basis functions, located along similar directions (i.e. basis functions that have narrow angles between them), the shortest weighted distance to the center is used as a secondary criterion to choose the basis function to be retained.

On termination, the above algorithm will return the k' nearest basis functions of $\phi_j(\cdot)$ in directions that are most widely separated from each other and subsequently each of them will be used to create a separate CSR for the basis function. The initial radius of each CSR is determined as half of the weighted distance to a neighboring basis center as

$$\tilde{\sigma}_l = \frac{1}{2}\sqrt{\sum_{i=1}^{d} \Theta_{ij}\left(\mu_{il} - \mu_{ij}\right)^2} \text{ for } l = 1, \cdots, k' \tag{4.37}$$

where $\boldsymbol{\mu}_j, \boldsymbol{\Theta}_j$ respectively are the center and the Discriminative Indices vector associated with the j^{th} basis function, k' is the number of CSRs in the basis function and $\tilde{\sigma}_l \in \{\tilde{\sigma}\}_j$ is the l^{th} CSR of the basis function in the direction of $\overrightarrow{\boldsymbol{\mu}_j \boldsymbol{\mu}_l}$.

4.6.3. Modifications to New Training Algorithms

The training algorithms outlined for the creation of DWRRBF networks can be used for CBF networks with minor modifications to handle multiple radii associated with each basis function. It is important to note that Radius Selection criterion Function is only an

algorithmic rule and therefore leads to minimal changes in the mathematical procedures used in the training algorithm. This is because a given input will always cause the selection of the same radius for a given geometry of basis centers. Therefore, the derivation of the training algorithm would view the basis function as having a single radius associated with the input (and other inputs in the same region of the basis space).

The weight update rules for Discriminative Indices $\{\boldsymbol{\Theta}\}$ in DWRRBF networks are given in equations (4.20) and (4.21). When these equations are applied to CBF networks, the Radius Selection criterion Function will affect only error surface gradient with respect to the Discriminative Index in (4.21). Therefore, by incorporating the RSF, the partial derivative is modified to

$$\frac{\partial E(\mathbf{X})}{\partial \Theta_{ij}} = -\frac{1}{N}\sum_{p=1}^{N}\sum_{l=1}^{q}\left(y_{lp} - t_{lp}\right) w_{lj} \exp\left(-\frac{\sum_{r=1}^{d}\Theta_{rj}\left(x_{rp} - \mu_{rj}\right)^2}{2\left(\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i\right)\right)^2} \right) \frac{\left(x_{ip} - \mu_{ij}\right)^2}{2\left(\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i\right)\right)^2} \qquad (4.38)$$

where $\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i\right)$ is the Radius Selection criterion Function defined in (4.29) to (4.31) and rest of the parameters are defined as for (4.21).

Because of the multiple radii associated with each basis function, the update rules (4.18) and the (4.19) for the overall function radius, σ_j of DWRRBF, need some modifications when applied to CBF network training. In a CBF network, a given radius $\tilde{\sigma}_l \in \{\tilde{\sigma}\}_j$ will be selected only for a subset of data points in the training set. Therefore, the updates for each of them should also be restricted to use the relevant subset of training data, thereby allowing the CSR to be determined according to local properties of the data set. As a result, in the batch-version of the gradient-descent algorithm which is used here, averaging in (4.19) must be made with respect to the set of training data for which the given radius is selected. From an

algorithmic point view, this can be achieved by associating a "hit counter" with each radius and incrementing its value each time the radius is selected.

By including the Radius Selection criterion Function in the gradient of the error surface with respect to the overall function radius described in (4.19), the modified gradient for CBF network is obtained as

$$\frac{\partial E(\mathbf{X})}{\partial \tilde{\sigma}_l} = \frac{1}{\tilde{N}_l} \sum_{p=1}^{N} \sum_{s=1}^{q} \left(y_{sp} - t_{sp} \right) w_{sj} \exp\left(- \frac{\sum_{i=1}^{d} \Theta_{ij} \left(x_{ip} - \mu_{ij} \right)^2}{2\left(\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i \right) \right)^2} \right) \frac{\sum_{r=1}^{d} \Theta_{rj} \left(x_{rp} - \mu_{rj} \right)^2}{\left(\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i \right) \right)^3} \quad (4.39)$$

where $\tilde{\sigma}_l \in \{\tilde{\sigma}\}_j$ is the l^{th} CSR of the j^{th} basis function, $\tilde{N}_l$ is the number of times that $\tilde{\sigma}_l$ is selected in one epoch, $\mathrm{Sel}\left(\{\tilde{\sigma}\}_j \mid \mathbf{x}_i \right)$ is the Radius Selection criterion Function and the rest of the parameters are as defined for (4.19). Apart from the above, no further modifications are required for the creation and training of CBF networks compared to DWRRBF networks.

4.7 Summary

In this chapter, a novel classification system for holistic facial expression recognition was developed based on RBF network architecture as the starting point. Holistic recognition of facial expressions is characterized mainly by a large-dimensional input space containing a significant amount of irrelevant variations, due to differences in faces among people. Furthermore, expression classes in the problem domain are described by different facial regions, which cause the input space to contain features that are important for the description of only some of the expression classes.

Statistical techniques that are commonly used for creation of RBF networks fail to deliver the best solution in the above conditions due to limitations like large number of learnable

network parameters and the lack of sufficient amount of training data. Therefore, a different approach is taken in the proposed solution to address these issues using a combination of statistical and algorithmic methods. In the proposed classifier, two new types of basis functions are introduced in order to improve the network's performance under the properties of the problem-domain. The first type of basis function (DWRRBF) introduces the concept of a weighted distance in a spherical basis function, which allows the emphasis of features that are important for the discrimination. The second type of basis functions (CBF) on other hand adds another level of non-linearity by further partitioning the input space into several intra-basis function segments using the local properties of the basis function. This partitioning allows the same basis function to contain multiple boundaries in the input space that are determined according to the separation from its surrounding neighbors.

In the following chapters details of a series of experiments that were carried out using the proposed classifiers will be discussed.

CHAPTER 5

A Facial Image Database and Test Datasets for Holistic Facial Expression Recognition

A major difficulty that is faced by many researchers for holistic recognition of facial expressions is the non-availability of a suitable test image database. Due to high dimensionalities involved, holistic approaches require fairly large amounts of data to train the classifiers compared to feature-based approaches to the problem. Most of the static facial image databases that are available have been created for different applications like face detection and face recognition. Therefore these databases contain images under varying conditions of background, lighting, shadows and pose. Even though some of these databases contain images of the same subject with different facial expressions, the variations are often restricted to few types of mixed expression classes.

We came across the same difficulty in obtaining a suitable image database and therefore the initial developments on the proposed algorithms were done using a temporary image database. The temporary database consisted of facial images collected from various sources [116][117][118] and showed only four different types of facial expressions that resembled Neutral, Angry, Smiling and Screaming faces. However, a more complete image database, containing all six classes of universal facial expressions was later created and was used in the experiments described in this thesis. The new database consisted of several facial images obtained from a database being developed at the Carnegie Mellon University (CMU) and a set of images photographed at the National University of Singapore (NUS). In the following

sections, the procedures followed in the normalization and the creation of datasets using this image database is presented.

5.1 Source Image Database

Facial images that were obtained from the database being created at CMU were not normalized to experiment with expression recognition. The database itself was intended for a slightly different purpose; for the recognition of FACS Action Units for subsequent analysis of facial expressions [119]. Moreover the version available was in its initial stages, consisting of medium quality, un-labeled and un-processed images. Therefore these images had to be processed and normalized extensively prior to their use in the classification experiments.

Figure 5.1: A sample of images created at NUS.

The images from the CMU database were 640×480 $(\mathrm{W} \times \mathrm{H})$ pixels in size (except for a minority which were of 640×490 pixels) and consisted of a complete frontal view of a human face displaying a facial expression. Variations in scaling were present and hence images of different subjects occupied an area of between 40%-70% of the total image size. In spite of the uniform lighting conditions, some images showed signs of gray-level saturation that could be attributed to incorrect white level setting in the digitization process. Saturation occurred mostly around cheek, forehead and chin areas of the facial image. Some variations in rotation and translation were also present among images of different subjects.

Images created at NUS too (Figure 5.1) followed a setup similar to that of the CMU database except for the image size, which was 1024×768 $(\mathrm{W} \times \mathrm{H})$ pixels. The frontal images were photographed using a digital camera mounted approximately at the height of the subject's mouth and about 1.5m away from the face. No special lighting (except for ambient fluorescent lighting in the laboratory) was used. However a uniform white screen was used for the background. Flash photography was not used since it created shadows in the facial image. Demographically, the subjects were mostly of south and east Asian origin whereas the subjects in CMU image database were mostly of Caucasian and African American origin.

5.1.1 Normalization of Facial Images

For holistic recognition of images, one of the primary requirements is their normalization and registration with respect to rotation, scaling and translation. Normalization eliminates unwanted variations arising from these parameters prior to classification. For facial images, other variations are present because of the structural difference in facial proportions that exist among different subjects, especially from different demographic groups. Differences in facial proportions affect the relative geometry of important facial component, such as the eyes, nose and the mouth region. Since variations in these regions are significant for the description of facial expressions, further normalization is required with respect to the relative placement of these features in the facial image. However, at the same time normalization should not affect the non-rigid deformation caused by facial expressions. Therefore, in order to strike a balance between the two conflicting requirements, normalization is done with respect to some static features that are least affected by the facial expressions.

Anatomically, all major components of the facial skeleton except the lower mandibles are attached rigidly to the skull. Thus, the mouth region becomes the most mobile facial area. On the other hand feature points like the centre of the eye-cavity (of the skull and not the

pupil centers) and the nose tip are subject to the least amount of deformation during a facial expression. Therefore, the latter can be used as reference points for the normalization for rigid image transformations and variations due to differences in facial proportions across different people in the image database. However, it must be noted that in practice an accurate estimation of these points in the facial skeleton may not be possible and therefore their approximate locations in the facial mask must be used instead.

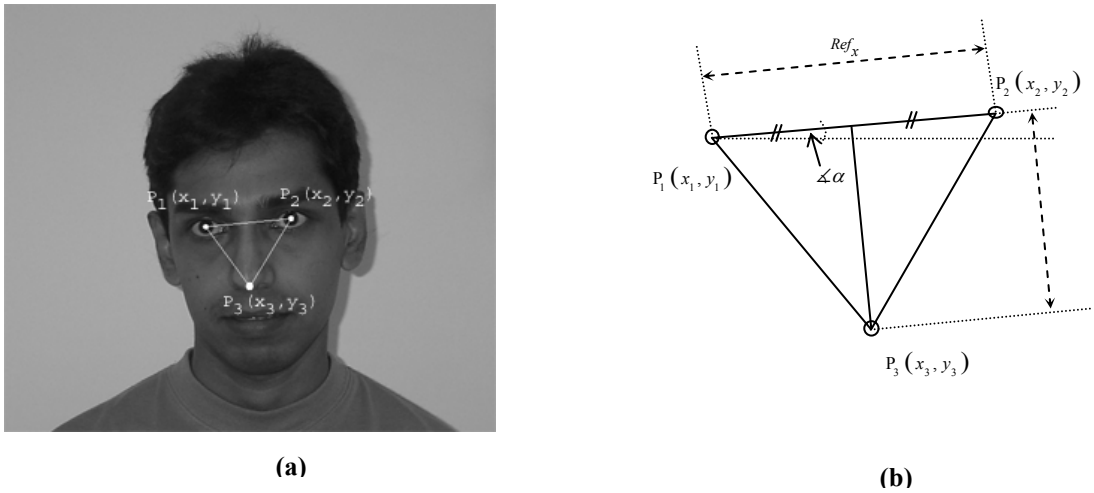


Figure 5.2: Reference points used in the normalization of facial images.

Based on the above considerations the normalization of facial images was done using three reference points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$ denoting the centre of the left eye cavity, center of right eye cavity and the nose tip (Figure 5.2a). For all images in the database, these points were marked manually by a mouse click using a specially developed GUI application. In order to minimize operator errors, the process was repeated three times (by different people) and the average coordinates were used. Once marking was completed for all the images, three basic measurements (Figure 5.2b) of facial geometry were computed for all images as follows:

$$\alpha = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{5.1}$$

Ref_y

$$Ref_x = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.2}$$

$$Ref_y = \sqrt{\left(\left(\frac{x_1 + x_2}{2}\right) - x_3\right)^2 + \left(\left(\frac{y_1 + y_2}{2}\right) - y_3\right)^2} \tag{5.3}$$

The first parameter α in (5.1) defines rotation of the image as the inclination of the eye-centers from the horizontal axis while the other two parameters provide overall width and height of the face. Statistics of these parameters, computed over the entire image set is given in Table 5.1. The results showed a significant variation in the two measurements Ref_x and Ref_y indicating wide variations in scaling. Likewise variations in the ratio of Ref_y / Ref_x further indicated the existence of considerable differences in facial proportions across different subjects in the database.

Parameter	α° (degrees)	Ref_x (pixels)	Ref_y (pixels)	Ref_y / Ref_x
Mean (μ)	-0.5253	100	60	0.5977
Std. Div. (σ)	2.6480	10	9	0.0972

Table 5.1: Statistics of facial proportions (before normalization) computed for all images in the database.

During normalization, the image was first rotated by the angle α in the clock-wise direction for rotation normalization. Thereafter using an affine transform, the image was scaled asymmetrically in the horizontal and vertical directions with scale factors S_x and S_y, respectively, computed as

$$S_X = \frac{40}{Ref_x} \tag{5.4}$$

$$S_Y = \frac{24}{Ref_y} \tag{5.5}$$

using reference measurements Ref_x and Ref_y of the image. For both operations a coordinate system with its origin at the nose-tip reference was used to compute pixel

coordinates while bi-cubic interpolation was used to determine their intensities when using the affine transform. The use of different scaling factors in the horizontal and vertical directions normalized Ref_x and Ref_y measurements, respectively, to their nominal values of 40 pixels and 24 pixels in all images. Additionally it also maintained a uniform facial proportion of 0.6 for the ratio of Ref_y / Ref_x , based on its average value in original images.

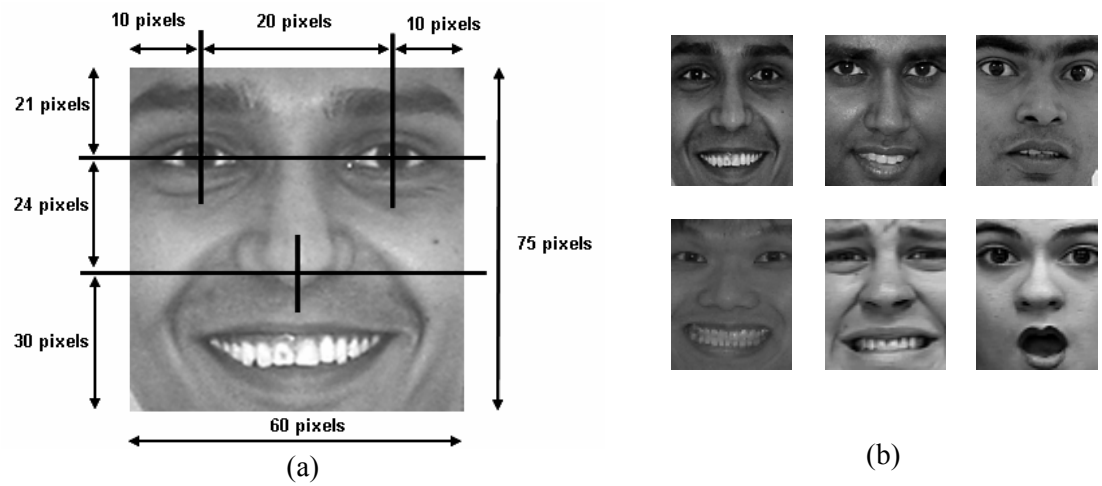


Figure 5.3: Cropped facial images. (a) Boundary details for image cropping. (b) A sample of cropped images in the database.

5.1.2 Image Clipping and Normalization for Average Intensity

Many researchers believe that the majority of information regarding facial expressions is concentrated into a few regions of the face such as the eyes, eye-brows and the mouth region [42][46]. In comparison, the outer face regions contain the majority of information regarding the subject's identity and therefore must be excluded from the holistic input when recognizing facial expressions. Based on the above observations, for experiments described in this thesis the facial images were cropped 60×75 pixels using the normalized images as illustrated in Figure 5.3. This retained the variation in facial expressions while minimizing the identity variations between individuals.

The cropped images enclosed all three facial regions that were considered primarily important for description of facial expressions. The relationship of cropping region to the

three normalized reference points was determined experimentally using a sample of 90 images picked randomly from the database. A comparatively larger height of $1.25 \times Ref_y$ was required below the nose tip reference in order to enclose wider variations of the mouth shape. However, it must be noted that for a minority of images it was not possible to include the complete mouth within this distance, especially with the "Surprise" expression. On the other hand it was also not possible to further extend the mouth region because this would then expose the chin-boundary and a portion of the neck in other types of expression where the mouth is closed. Nevertheless, test results later proved that partial inclusion of the fully opened mouth was sufficient for recognition of the expressions.

Since the images were obtained from two difference sources (i.e., CMU database and those photographed at NUS), a considerable variations were observed in their gray level distributions. Additionally the 8-bit pixel depth was likely to result in numerically large distance measures within the classifier, which in turn posed some danger of numerical problems. Due to these reasons, the gray level values in the images were first scaled to the range from 0 to 1 and subsequently normalized again for zero mean and unit variance.

5.2 Creation of Training / Test Datasets

Using images normalized as described above, three different versions of datasets were created as described below:

1. **Primary dataset** that consisted of the 60×75 pixels normalized facial region as described in the previous section.
2. **Expression Feature Region (EFR)** that consisted of pixels values representing facial regions of eyes, eye-brows and the mouth.
3. **Half-face** dataset that consisted of left half of the facial region used in the Primary dataset.

Each of the three datasets consisted of 411 images belonging to a set of 98 subjects. Moreover, 22 out of the 98 test subjects had images of all six expressions included in the image database.

From the above three datasets, the primary dataset consisting of the complete 60×75 facial region was used as the main source of image data in all experiments described in this thesis. Since a majority of classifiers required their input to be in a vector, each image in the dataset was converted into a 4500×1 dimensional vector by column concatenation. The EFR and half-face datasets on the other hand were used in some of the supplementary experiments that evaluated the performance of the proposed classifiers on rather low dimensional spaces compared to the Primary dataset.

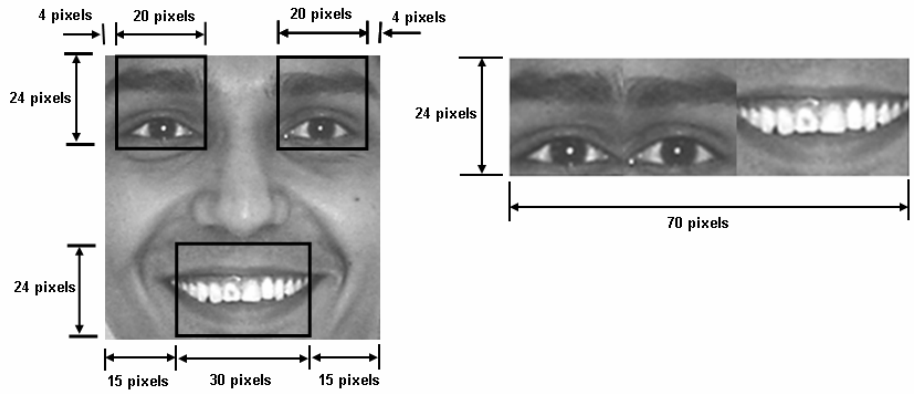


Figure 5.4: Composition of Expression Feature Regions (EFR) dataset.

Details of facial regions used in the creation of EFR dataset are illustrated in Figure 5.4. Facial regions similar to these have also been used previously by Kobayashi and Hara [46] in their Facial Characteristic Points (FCP)-based approach to facial expression recognition. The FCP approach suggested that the height of the eye-plane (from bottom of the lower lip to the top of eye brows) and the height of the mouth region (from the bottom of the lower lip to the top of the upper lip) to be similar and numerically around 90% of the distance between the eye (pupil) centers. However, in an early investigation, it was discovered by us that mainly due to the normalization of images, a height of 60% of the eye-distance was

sufficient. Considering the above facts and the dimensions of the normalized image, as illustrated in Figure 5.4, the eyes were extracted as 20×24 pixel regions each. Similarly, the mouth was extracted as a 30×24 pixel region from bottom of the normalized image, also considering the fact that anatomically the typical human face is divided into two equal halves by the horizontal line passing through the tip of the nose [120].

In order to create the EFR dataset the respective facial regions (Figure 5.4) were first extracted as two 24×20 pixel image segments from the eye region and a 24×30 pixel image segment from the mouth region of a normalized facial image. Next the three image segments were combined to form a 24×70 pixel image, which was finally converted into a 1680×1 dimensional vector using column concatenation.

The half-face dataset was created in the belief that the human face is symmetrical across the vertical line passing through center of the nose. Consequently this dataset was created using the 30×75 pixel region from the left half of the images included in the primary dataset. Similar to other two datasets, each image segment was converted to a column vector of dimension 2250×1 using column concatenation.

5.3 Summary

In this chapter detail of creation of a primary facial image database and other datasets was presented. The image database was created using a section of facial images obtained from the CMU face database and some images created at NUS. After normalizing these images for their variations in rigid deformations, average intensity and some differences in facial proportions of different people, three training / test data sets were created. Each dataset was created based on a different region of interest, extracted from images in the source database.

CHAPTER 6

Results and Discussion

In order to evaluate the performance of proposed algorithms for the holistic recognition of facial expressions, a series of experiments were designed using training/test datasets described in the previous chapter. These experiments included tests using the new algorithms that were proposed in Chapter 4 as well as tests using some of the traditional RBF network-based techniques that were discussed earlier in Chapters 2 and 3. In this chapter, experimental details and results are presented. In Section 6.1, details of training and validation procedures are described. In Sections 6.2 and 6.3 results obtained with the new DWRRBF and CBF networks, respectively, using the primary dataset are discussed. Results of experiments on EFR and Half-face datasets are presented in Section 6.4. Experimental results for other types of RBF network-based classifiers and those using dimensionality reduction methods are discussed in Sections 6.5 and 6.6, respectively. Finally, in Section 6.7 a comparison between proposed techniques and the other types of RBF networks is presented.

6.1 Training and Validation Datasets

Because of the limited amount of image samples that were available in the dataset, a cross-validation method was used. Here, a portion of the dataset is left out during the training procedure and is used as a test input for validation. First the entire dataset (411 images) was divided into five subsets (Subset A to Subset E) of approximately 80 images each, selected randomly. The composition of these subsets is shown in Table 6.1. Even though images

were selected individually from each expression-class, no specific effort was made to include images of the same person in a single subset.

Expression Class	Number of images in each subset					Total number of Images
	Subset A	Subset B	Subset C	Subset D	Subset E	
Fear	13	13	13	13	14	66
Surprise	19	19	19	19	19	95
Sad	15	15	15	15	14	74
Angry	8	8	8	8	8	40
Disgust	10	10	10	10	9	49
Happy	17	17	17	18	18	87

Table 6.1: Composition of expression classes in the 5 data subsets.

(i) (ii) (iii) (iv) (v) (vi)

Figure 6.1: Typical images in the database. (i) Fear, (ii) Surprise, (iii) Sad, (iv) Angry, (v) Disgust and (vi) Happy.

For each training cycle, four out of five subsets was used for training and the fifth subset was reserved as the validation test-set. The procedure was repeated for all five data subsets of data and their averaged outcome was taken as the final result.

6.2 Performance of the Differentially Weighted Radius Radial Basis Function Network

In this section, results obtained using the Differentially Weighted Radius Radial Basis Function (DWRRBF) network, which was proposed earlier in Section 4.2, are discussed. The summary of the overall performance, using a single non-hierarchical DWRRBF network is shown in Table 6.2. The best average recognition rate of 84.9% was recorded when Discriminative Indices used in the DWRRBF network were computed according to the mean criterion (4.9). In contrast, when the variance based criterion (4.8) was used to compute

Discriminative Indices the performance was slightly degraded with an average recognition rate of 79.8%. Both networks however, showed similar trends in recognition of the individual expression classes. The best accuracy was obtained for the Surprise expression followed by the Happy expression. On the other hand, the lowest recognition rate in both cases was recorded for the Fear expression class.

	Accuracy of expression recognition						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
Using Discriminative Indices computed with variance criterion (4.8)	51.5%	92.6%	79.7%	65.0%	85.7%	90.8%	79.8%
Using Discriminative Indices computed with mean criterion(4.9)	63.5%	94.7%	81.1%	77.5%	89.8%	94.2%	84.9%

Table 6.2: Results for DWRRBF network with non-hierarchical classification (with 44 basis functions in the network).

The difference in the overall recognition rates of the two criteria (Section 4.2.2) used to compute the Discriminative Indices can be explained using their responsiveness to global and local variations of features. Between the two methods the variance-based criteria (4.8) is more biased towards global distribution of features in input space. This is because of the presence of total variance component and due to the averaging of variances in expressions over different individuals in the numerator of (4.8). On the other hand the mean-based criterion (4.9) is more responsive of the localized separation of features compared to the variance-based criterion. Discriminative Indices computed using (4.9) are subject to a smaller degree of averaging since the computation of the separation between basis functions (i.e. the numerator in (4.9)) is limited to the $\tilde{h}$ basis functions in the local neighborhood.

Additionally the denominator of (4.9) represents only the variances within the local region represented by the basis function. Therefore the mean-based criterion is more responsive to the local variations of features, which in turns can be related to differences in facial expressions compared to structural differences which lead variations to spread over the entire data set.

Using an argument similar to the above, the relatively inferior recognition rates of Fear, Anger and Sad expressions can be explained as follows. These expressions differ significantly in the eye and eye-brow regions of the face. These regions have a relatively smaller degree of variations in pixel values due to (all classes of) facial expressions as compared to variations arising from different individuals. Compared to the eye and the eye-brow regions, the mouth and the inner-cheek regions of the face have wider variation in pixel values arising from different facial expressions as opposed to the variations caused by structural differences in individual faces. The larger variations in the mouth and inner cheek regions tend to dominate the Discriminative Indices vector, thereby influencing the weighted distance, computed for the basis function, to be biased more towards features belonging to these regions.

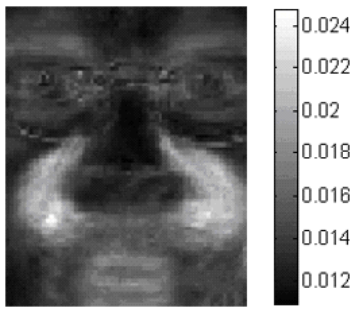


Figure 6.2: Discriminative indices computed using the variance criterion (4.8).

The dominance of Discriminative Indices corresponding to mouth and the inner-cheek regions is clearly evident in the image of Figure 6.2, which shows the initial values of Discriminative Indices computed using the variance criterion (4.8). Note that the variance

criterion computes only a single initial vector of Discriminative Indices for all basis functions. Concentrations of higher valued Discriminative Indices in the mouth and the inner-cheek regions leads to a better recognition of expressions such as Happy, Surprise and Disgust that are separated mostly by the mouth shape and wrinkle in the inner cheek region compared to other expressions that are separated by features in the eye and the eye-brow regions.

6.2.1 A Hierarchical Structure for Classification

One of the best methods to address the above problem of dominant feature variation is to structure the classification hierarchically, where grouping of expression classes at each level is dominated by their locally important properties. This approach was first proposed by Daw-Tung et. al. [58] who suggested a two-level hierarchy with a structure determined according to visual cues in facial expressions. Classification in the first level of their method was based on variations in the mouth region while at the second level, features in eye regions played an important role. This division, however, was not very successful as their results showed poor recognition in the second level compared to the first. Therefore, in the hierarchical DWWRBF network proposed here the structure was determined experimentally, based on its performance on a non-hierarchical classifier using a random subset of training samples.

		Non hierarchical DWRBBF Network output					
		Fear	Surprise	Sad	Angry	Disgust	Happy
Class label	Fear	**21**	2	1	3	1	12
	Surprise	0	**37**	2	0	0	1
	Sad	0	0	**28**	6	6	0
	Anger	1	0	3	**32**	0	2
	Disgust	1	0	3	3	**32**	1
	Happy	5	0	0	2	1	**32**

Table 6.3: Confusion matrix for a random sample of 240 images, using Discriminative Indices computed according to variance criterion (4.8).

The confusion matrix returned by the non-hierarchical classification of 240 images (40 images in each expression class) using DWRRBF network with Discriminative Indices computed according to the variance criterion is illustrated Table 6.3. The experiment used an equal number of images in all classes in order to avoid any effects due to unequal sample size for different classes. Also the variance criterion was selected for this experiment because it is more representative of the averaging of locally important features and the effects of globally distributed features, which lead to the inferior recognition trends in Table 6.2. Similar to the results shown in Table 6.2, the experiment returned the lowest recognition rate for Fear expression with only 21 out of 40 images in the group being recognized correctly. Within misclassified set of images, 12 images belonging to Fear expression were recognized incorrectly as Happy while 5 images belonging to Happy expression were misclassified as Fear. Thus, the highest number of confusions occurred between Fear and Happy expressions with 17 out of 80 images (21.2%) in both classes being misclassified into each other. Apart from these two categories, the second highest amount of confusion was recorded in relation to the Sad expression. There were 9 confusions each recorded between Angry vs. Sad and Disgust vs. Sad expressions in addition to another 3 confusions between Angry and Disgust expressions. As a result, a total of 21 out of 120 (17.5%) confusions were recorded within the three categories.

Based on the above results, the expression categories for the first level of classification were determined as illustrated in Figure 6.3, by combining expression classes with the most number of confusions in a single category. Consequently, according to results presented in Table 6.3, Fear and Happy expressions with the largest number of confusions (21.2%) were combined into “Category A”. Similarly, “Category C” was formed by combining Sad, Angry and Disgust expressions with 17.5% confusions among them. The Surprise expression, which caused the least amount of confusion, was left in its own “Category B” in the first level of the hierarchy.

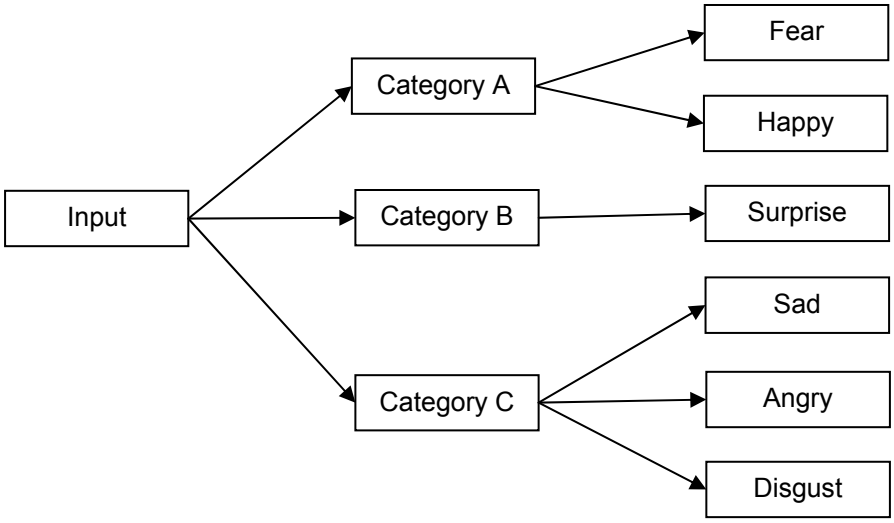


Figure 6.3: Two level hierarchical classification structure.

With the above hierarchy defined, the complete classification system consisted of three DWRRBF networks. One network provided the first level categorization while the other two further discriminated "Category A" and "Category C" images into their respective expression classes at the second level (Figure 6.3). All the three networks were trained independently with their own sets of Discriminative Indices computed using images in their respective combined pattern classes. The effects of classification hierarchy are clearly visible in images of respective Discriminative Indices as illustrated in Figure 6.4. For example, the Discriminative Indices used in the first level of categorization (Figure 6.4a) have their dominant values distributed more evenly in all expression feature regions compared to the non-hierarchical approach which was illustrated earlier in Figure 6.2. Furthermore local differences in the mouth and eye-brow regions are better emphasized in the hierarchical approach (Figure 6.3a) than its non-hierarchical counterpart.

In contrast to the first level, the Discriminative Indices of the second level of classification show the emphasis on locally important features that are prominent within their respective expression classes. For instance, those associated with the separation between Fear and Happy expressions (Figure 6.4b) have the emphasis on mouth, upper eye-lid and the eye-brow region. According to the Facial Action Coding System (FACS) [8][119] the

(a) (b) (c)

Figure 6.4: Images of initial Discriminative Indices (computed using (4.8)) in a hierarchical classification structure. (a). First level with three combined classes, Category A, Category B and Category C. (b). For separation between Fear and Happy at second level. (c). For separation among Sad, Angry and Disgust at second level.

main differences in these expressions include the raised eye-brows (AU1+AU4 according to FACS Action Units) and raised upper eye-lids (AU5) for Fear expression and wide opened mouth (AU12+AU16+AU26) and raised cheeks (AU6) for Happy expression. In the same way, Discriminative Indices for separation of Sad, Angry and Disgust expressions in "Category C" are distributed more evenly over more facial regions with some emphasis on the narrow mouth region, eye region and the region of nose wrinkles. Of the three expressions Sad and Angry in general have somewhat similar characteristics in the mouth region. According to the FACS descriptors, one major difference between the two expressions belonging to the eye region with the presence of AU1, which is defined as raised inner eye-brows in Sad expression. Both expressions on the other hand exhibit a closed and narrowed mouth shape with Angry expression having tightly pressed lips (AU24 + AU23) compared to the Sad expression. However, it must be noted that differences in pixel values, especially in the eye region caused by these deformations are of less significance compared to those in the mouth region for other types of expressions. The Disgust expression, compared to Sad and Angry expressions, on the other hand, creates a larger deformation in the inner cheek region (AU9 + AU6+AU10) which corresponds to nose wrinkles and a raised inner cheek/upper lip region. The significance in these differences of the inner cheek area are captured clearly in Figure 6.4c with a small concentration of higher-valued Discriminative Indices around the nose and inner cheek regions.

	Accuracy of expression recognition in hierarchical DWRRBF network						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
Discriminative Indices computed using variance criterion (4.8)	87.9%	93.7	95.9%	92.5%	89.8%	94.2%	92.7%
Discriminative Indices computed using mean criterion(4.9)	84.8%	94.7%	94.6%	90.0%	87.8%	96.5%	92.2%

Table 6.4: Overall results for 2-level hierarchical classification with DWWRBF networks.

6.2.2 Performance of Hierarchical Classification

The overall performance of the two level hierarchical classification system is presented in Table 6.4. It must be noted that, even though only a subset of 240 images were used to determine the class hierarchy, training of respective DWRRBF networks was carried using the complete dataset according to cross-validation criteria described in Section 6.1. The results showed significant improvement in performance with an overall recognition rate of 92.7% compared with the best of 84.9% in the non-hierarchical configuration. The improvement was contributed mainly by the Fear expression (87.9%), which recorded the lowest recognition rate in the non-hierarchical method. Moreover, Sad and Angry expressions also demonstrated significant gains which is attributed to the better emphasis on their local variations in the hierarchical structure. Additionally, the hierarchical classifier showed compatible levels of performance in both criteria for computation of Discriminative Indices (Table 6.4). This can be attributed to the fact that the bias towards local properties in the means criterion (4.9) becomes less significant in the hierarchical classification, because of the merger of expression classes with similar local properties.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**87.9%**	-	9.1%	-	1.5%	1.5%
	Surprise	4.2%	**93.7%**	2.1%	-	-	-
	Sad	2.7%	-	**95.9%**	1.3%	-	-
	Anger	2.5%	2.5%	-	**92.5%**	2.5%	-
	Disgust	2.0%	-	2.0%	2.0%	**89.8%**	4.0%
	Happy	-	-	1.1%	1.1%	3.4%	**94.2%**

Table 6.5: Overall confusion matrix for two level hierarchical classifier using Discriminative Indices computed according to variance criterion (4.8).

		Network output		
		Category A	Category B	Category C
Class label	Category A	**92.2%**	-	7.8%
	Category B	4.2%	**93.7%**	2.1%
	Category C	3.7%	0.6%	**95.7%**

Table 6.6a: Confusion matrix for first level of classification.

		Network output	
		Fear	Happy
Class label	Fear	**98.5%**	1.5%
	Happy	2.3%	**97.7%**

Table 6.6b: Confusion matrix for second level of classification of Category A.

		Network output		
		Sad	Angry	Disgust
Class label	Sad	**98.6%**	1.3%	-
	Angry	2.5%	**95.0%**	2.5%
	Disgust	2.0%	2.0%	**95.9%**

Table 6.6c: Confusion matrix for second level of classification of Category C.

The overall confusion matrix for the two level classification system is illustrated in Table 6.5 whereas Table 6.6a, Table 6.6b and Table 6.6c, respectively show the confusions that occurred in individual networks at first and second levels of the classification system. In all the networks, the respective Discriminative Indices were computed using the variance criterion according to (4.8). The results show that most of the confusions have occurred in the first level of the classifier, where the discrimination was more biased by the concentration of feature variations in the mouth and the inner-cheek regions (Figure 6.4a). However, unlike variations in the eye region, variations in these areas are not only

contributed by the facial expressions but also by some major differences originating from identity information among different test subjects. Often, the same pixel sub-region is subject to combined variations due to both the facial expressions as well as the identity information. This has made extraction of the required information a difficult task, even with the hierarchical structure.

6.2.3. Recognition Rate and Dimensionality of the Basis Space

The iterative algorithm for creation of a DWRRBF network described earlier in Section 4.3.1, starts with only a single basis function to represent each category of facial expression included in the training dataset. New basis functions are then added to the network in each iteration, based on training images that are misclassified by current configuration of the network. An example of the performance of the DWRRBF network, according to the number of basis functions included after each cycle of the iterative algorithm is illustrated in Figure 6.5. The data in Figure 6.5 was extracted from the first level-classifier used in the hierarchical classifier described in the previous section.

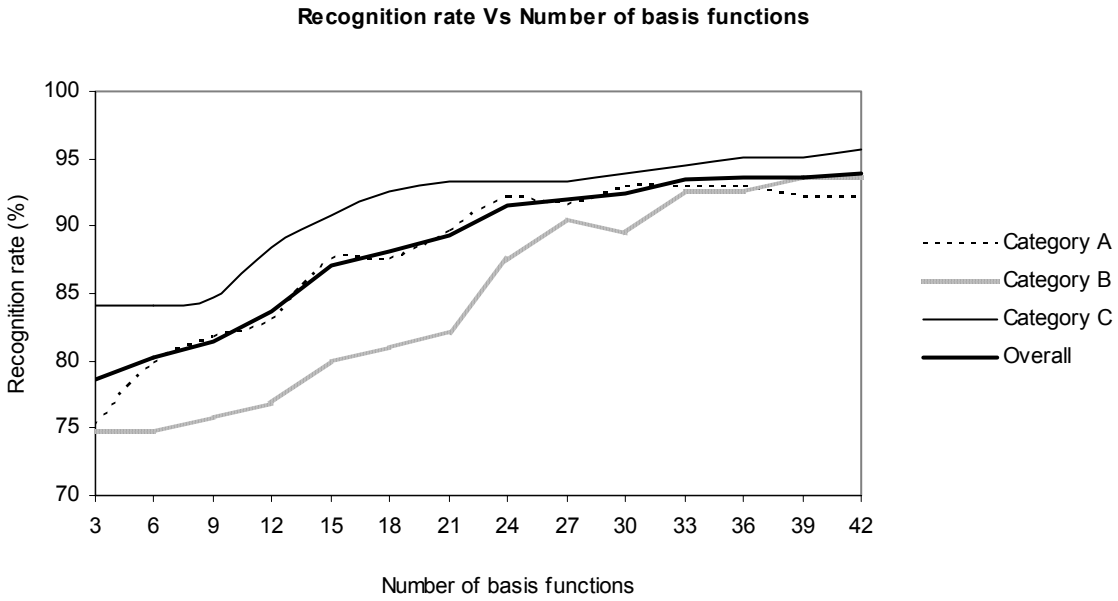


Figure 6.5: Variation of the network performance against number of basis functions in the network for first level of the hierarchical classifier.

The results in Figure 6.5, show that the overall performance of the network increased rapidly with the addition of new basis functions during the early stages of the training process. Later, the recognition rate started to saturate around 95% with more than 33 basis functions included in the network. This is reasonable since at this stage the network has sufficient number of basis functions to represent all major homogeneous clusters in the training data. As a result the addition of new basis functions will be used for representing isolated samples and therefore will not improve the general representation of input space by the network. If the above procedure is allowed to continue, new basis functions will be added until eventually all outliers in the training set are included. The net result of this will be to compromise the generalization of the network because additional basis functions in the network will also affect major cluster boundaries represented by other basis functions. Indication of this phenomenon is evident in Figure 6.5 where "Category A" shows signs of decreasing recognition rate beyond a 33 basis functions. Therefore, when designing a DWRRBF network a compromise must always be made on the number of basis functions against the expected performance. This, in general, can be achieved by specifying an appropriate minimum cluster size for creation of new basis functions in the training algorithm described in Chapter 4.

6.2.4 Parameter Learning in DWRRBF Networks

Apart from linear weights in the post-basis mapping, there are two other types of learnable parameters in a typical DWRRBF network. These include Discriminative Indices $(\boldsymbol{\Theta})$ and the overall basis function radius (σ). However, unlike the post-basis mapping, initial values of these parameters of the network are computed according to some statistical properties in the training dataset. Subsequently, in order to allow these parameters to better adapt to their local environments further optimization is performed using a supervised gradient-descent training algorithm.

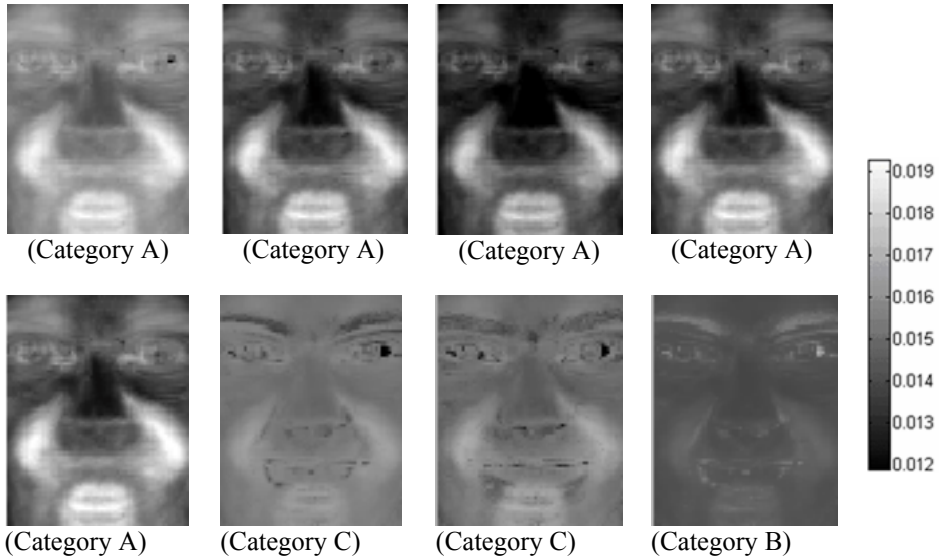


Figure 6.6: A sample of Discriminative Indices associated with different basis functions in the first level of the hierarchical classifier after the gradient descent training algorithm has converged. Shown below each image is the class represented by their respective basis functions.

(a). The Discriminative Indices: A sample of Discriminative Indices in the first level classifier of the hierarchical classification system is illustrated as images in Figure 6.6. Note that Discriminative Indices in this example were computed using the variance criterion (4.8) and hence at beginning of the training procedure all were initialized to the same vector, similar to that illustrated in Figure 6.4a. Thereafter, the values were iteratively updated in the direction of negative gradient of error surface according to (4.20). While changes in most of the basis function were rather small, Figure 6.6 clearly shows that some basis functions have adapted significantly. For instance, the differences in the overall intensity of Discriminative Indices vectors representing basis functions of the same category (e.g. "Category A") suggest that they have adopted themselves for differences in the overall intensity of images representing the same class of data. Similarly, Discriminative Indices representing "Category B" and "Category C" expressions have changed with less emphasis on the mouth and inner-cheek regions compared to their counterparts representing "Category A" expressions.

(b). The Basis Function Radius: Parameter σ in (4.7) provides smoothing and plays an important role in performance of the network by determining an overall radius for the region represented by the basis function in input space. A shorter radius causes the basis functions to be less responsive to some data points in the cluster thereby incompletely mapping the input onto basis space. A larger radius on the other hand causes the responsive regions of different basis functions to overlap significantly thereby creating a more correlated basis space. The most common practice in determining this parameter in a typical RBF network is to use some heuristic rule, for instance, as a multiple of average or minimum distance between k nearest basis centers in a local neighborhood [20][82]. However, using a such simple heuristic rules becomes less practical in DWRRBF networks because each basis function will compute the distance of data points weighted by its own set of Discriminative Indices. Therefore, in the proposed network, the basis function radius is further fine-tuned together with the other parameters within the same gradient descent learning algorithm.

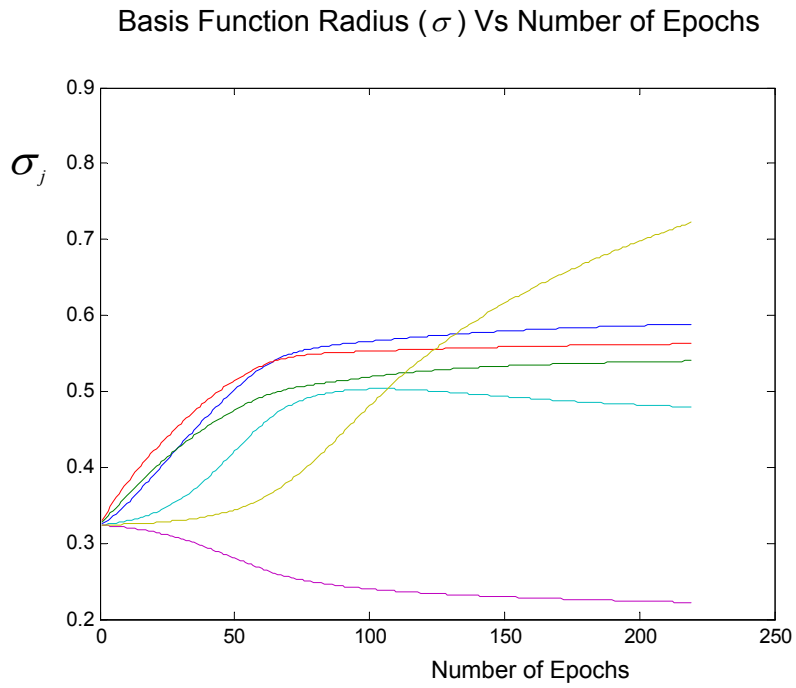


Figure 6.7: Learning the radius of different basis functions during the gradient descent learning algorithm.

Figure 6.7 shows typical learning characteristics of the smoothing parameter (σ_j) of different basis functions during the gradient-descent learning procedure. For clarity, only radii of 6 randomly selected basis functions are shown. Furthermore, the initial values for σ in all basis functions of the above example were set heuristically as the average distances between the six basis centers. The figure clearly demonstrates that the radius of a majority of the basis functions has converged within first half (i.e. within about 80 epochs) of the training procedure. The single basis function with increasing radius was found to be responding to a pattern class with features having a larger within-class variance compared to the rest. As shown by Moody and Darken [81], if the updates are continued for a sufficiently long period, a basis function like this would eventually evolve to cover a larger portion of the input space by attempting to include all data of that class within the region represented by the basis function. Moreover, the larger basis function is likely to overlap with basis functions representing other classes of data compromising the generalization properties of the network. This can however be prevented by stopping the gradient-descent algorithm at the appropriate time (Section 4.3.3), thereby allowing additional basis functions to be created by splitting the large basis function into multiple regions that are represented by different basis functions.

6.3 Performance of Cloud Basis Functions

In this section classification results obtained by using the Cloud Basis Function (CBF) networks proposed in Section 4.6 are presented and discussed. The CBF network differs from DWRRBF networks mainly by the fact that each CBF is associated with multiple values of radii compared to the single radius in the DWRRBF. These multiple radii represent segments of cluster boundaries separating each of its k nearest basis functions in the local neighborhood. Each value of radius in a CBF is referred to as a Cloud Segment Radius (CSR) whereas the boundary segment represented by the radius is referred to as the Cloud Segment (CS). As a result, in addition to the usual network parameters a CBF is also

characterized by the number of Cloud Segments that are present in the basis function. For a given input, the appropriate CSR (4.29) is selected, based on orientation of the test input with respect to some reference vectors representing direction of Cloud Segment. Because of multiple boundary segments, Cloud Basis Functions are capable of representing the skewed nature of local data distribution more accurately than DWRRBF.

	Accuracy of expression recognition in CBF network						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
Discriminative Indices computed using variance criterion (4.8)	92.4%	98.9%	95.9%	92.5%	91.8%	98.8%	95.9%
Discriminative Indices computed using mean criterion(4.9)	93.9%	98.9%	95.9%	92.5%	91.8%	98.8%	96.1%

Table 6.7: Results for Cloud basis function network with non-hierarchical classification. The network consisted of 9 basis functions, each having 4 Cloud segments.

Classification results obtained from the CBF network are given in Table 6.7. The network used in this experiment consisted of a non-hierarchical classifier with 4 Cloud Segments attached to each basis function. The results demonstrated a significant improvement in performance compared with results obtained for the DWRRBF network counterpart. There was nearly 11% increase in the overall recognition rate compared with the DWRRBF network using non-hierarchical classification (Table 6.2). On the other hand compared with the hierarchical classification system (Table 6.4), the CBF network demonstrated a 3.9% improvement in the overall recognition rate.

Another significant difference between the two network types was that the CBF network shows little dependence on the two criteria used in computing the Discriminative Indices. In the DWRRBF network with a non-hierarchical classification there was a difference of 5.1% in overall recognition rate for the two criteria (equations 4.8 and 4.9) of Discriminative

Indices. In contrast the CBF network yielded identical recognition rates for all the expression-classes except Fear, in which the difference was only 1.5%. This property of the CBF network can be attributed to its ability in representing different extents of separations within the same basis function, according to the local properties in the neighborhood. When neighbors around a basis function are separated with different *distances*, a CBF is capable of representing them using different CSR's whereas, in the DWRRBF, the radius will converge according to the shortest weighted-distance.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**92.4**%	-	3.0%	1.5%	1.5%	1.5%
	Surprise	1.0%	**98.9**%	-	-	-	-
	Sad	1.3%	-	**95.9**%	2.7%	-	-
	Anger	2.5%	-	5.0%	**92.5**%	-	-
	Disgust	4.0%	-	2.0%	-	**91.8**%	2.0%
	Happy	1.1%	-	-	-	-	**98.8**%

Table 6.8: Confusion matrix for non-hierarchical CBF classifier

The confusion matrix for CBF network is illustrated in Table 6.8. Although the number of confusions was small in the CBF network, confusions in individual expression classes followed a similar pattern as in the hierarchical DWRRBF network (Table 6.5). The highest amount of confusion was related to Fear expression with 7.6% of its images being confused with other classes of expressions. Conversely, between 1.0% and 4.0% of the other expressions were also misclassified as Fear. The lowest number of confusions was related to Surprise expression. Only a single image (1.0%) showing Surprise expression was misclassified as belonging to the Fear class.

6.3.1 Parameter Learning in Cloud Basis Functions

Compared to DWWRBF networks, learning in CBF networks is further characterized by two additional parameters; the multiple values of CSRs and reference vectors that define the orientation of their respective Cloud Segments. Since each Cloud Segment represents a

separation from one or more neighbors towards one direction of the basis function, the latter are determined according to the geometry (i.e. relative positions) of basis centers in input space. The geometry of basis centers is not affected by the gradient-descent algorithm and therefore the reference vectors (Section 4.6.1) defining respective Cloud Segments need not be modified further. Samples of such reference vectors related to CSRs that separated a basis function representing the Fear expression from its neighbors are illustrated in Figure 6.8. It must be noted that in this experiment, CSR's were computed using the 4-nearest neighbors and therefore there are only four reference vectors per each basis function. The emphasis on different facial regions in these images (Figure 6.8) clearly indicates that expression classes are not separated on similar directions within the input space.

Figure 6.8: Images showing four Cloud Segments in a CBF representing the Fear expression.

The distribution of CSR values for 6 randomly selected basis functions (representing each of the 6 expression classes) in the CBF network described in the previous section is illustrated in Figure 6.9. At the beginning of the gradient-descent learning procedure, initial values of all four CSR were set according to (4.37). After the learning algorithm had converged, the selected basis function representing the Fear expression was found to have the most variations in its CSR values. This can be attributed to the fact that this basis function is separated from its neighbors in varying extents compared to basis functions of other expression classes. The same observation also explains a reason for the expression's poor recognition rate, especially with the non-hierarchical DWRRBF network. When there is only a single radius, DWWRBF tends to converge at the shortest radius to avoid an overlap with its nearest neighbor. As a result only the portions of data within this shortest radius are

mapped correctly onto the basis space, resulting in a poor representation of the expression class in input space.

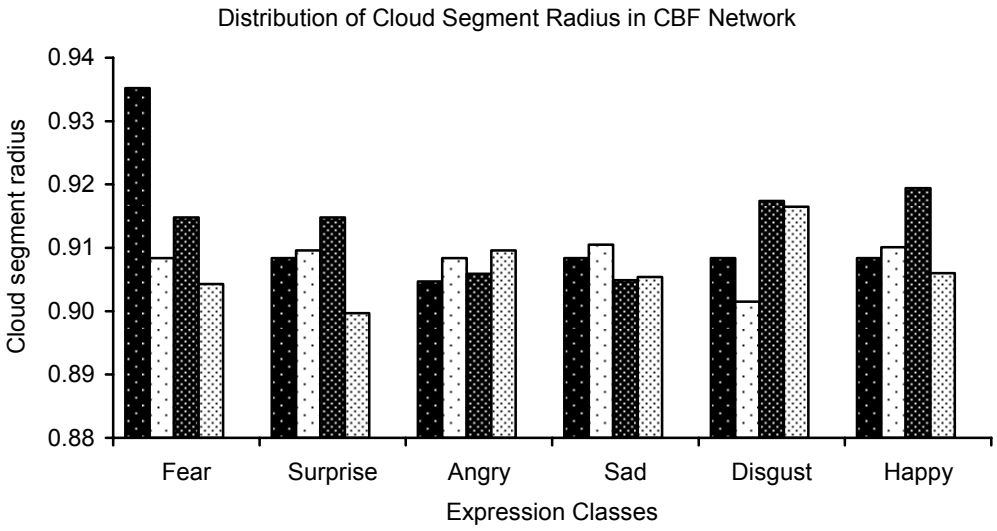


Figure 6.9: Distribution of CSR for each basis function in the CBF network.

6.3.2 Finding Optimal Number of Cloud Segments per Basis Function

Another important decision that must be made when developing a CBF network is the number of Cloud Segments to be included in each basis function. For a CBF layer with h nodes there can be as many as $h-1$ Cloud Segments associated with each basis function. Dividing a basis function boundary into higher number of Cloud Segments adds more versatility to the network but only at the expense of having even smaller amounts of data to train their parameters. Because of the Radius Selectivity Function in (4.29), only a single CSR is updated for any single training input. For that reason, unless every Cloud Segment in the CBF belongs to a direction with sufficient number of training data, some of its CSR's will not be properly trained during the gradient descent learning process. Additionally using separate Cloud Segments for neighbors that are separated in the same direction relative to a basis function will not bring any additional benefits since all their CSR's will converge to the same value, according to the nearest among them. Consequently, for the best performance, orientation of different Cloud Segments in a CBF must be distributed evenly in different directions based on its neighboring basis functions.

Some additional experiments using different number of expression classes showed that performance of a CBF network tended to peak with $c-1$ Cloud Segments per basis function where c is the number of pattern classes. This observation can be explained using the fact that with the weighted distance measure of Discriminative Indices, data clusters of the same class are more likely to be located in similar directions in input space with irrelevant variations only affecting their positioning relative to each other. Hence, all Cloud Segments separating data clusters of the same class will be oriented in similar directions and, as a result, their respective CSRs will converge to the same value, corresponding to the nearest neighbor among them.

The graph in Figure 6.10 demonstrates overall recognition rates obtained in experiments with CBF networks using different number of Cloud Segments per basis function. Note that when there is only a single Cloud Segment for each basis function, the network becomes identical to a DWRRBF network. Consequently, the performance obtained under this was nearly identical to that of the non-hierarchical DWRRBF network which was discussed earlier in Section 6.2.

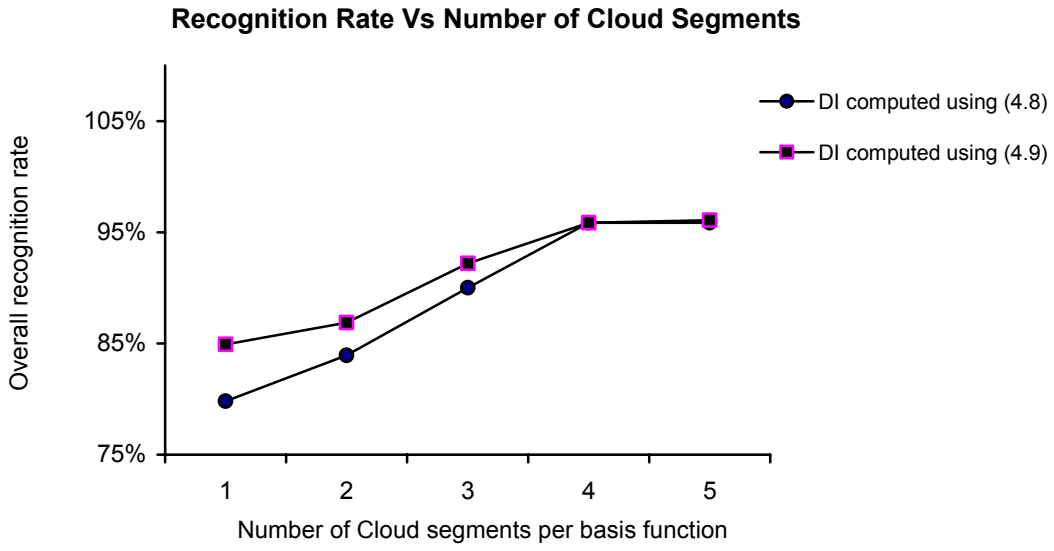


Figure 6.10: The overall recognition rate for two criteria of Discriminative Indices vs number of Cloud Segments per basis function in CBF network.

The CBF network for 6-expression classes showed signs of convergence with only 4 Cloud Segments associated with each basis function. A closer examination of the hyper-angles (4.30) between reference vectors that defined the orientation of these Cloud Segments revealed that the two expression classes Angry and Sad were oriented in similar directions with respect to other basis functions in the input space due to their similarity compared to other expressions, and the narrow separation between them in the eye and mouth regions. Hence, both classes contributed to the same boundary in separating them from basis functions of other expressions and as a result only a small gain in the performance was observed when two different Cloud Segments were assigned for these two classes compared to a single Cloud Segment.

6.3.3 A Comparison of CBF Networks and DWRRBF Networks

Table 6.9 shows a summary of operating parameters and results obtained for the two types of networks in experiments described in Sections 6.2 and 6.3. The best overall recognition rate of 96.1% was produced by the CBF network with four Cloud Segments for each basis function. Furthermore, this CBF network required only 9 CBFs in the hidden layer compared to 44 basis functions in the non-hierarchical DWRRBF network and a total of 85 basis functions in the three network hierarchical classification system. Three classes of expressions namely, Fear, Sad and Angry had two CBFs representing each of them in the hidden layer while the other three classes of expression were represented by a single CBF each. However it must be noted that the lower number of basis functions in a CBF network is achieved at the expense of more parameters being stored in each of the nodes. Each Cloud Segment in a CBF requires two additional parameters: a scalar value for CSR and a reference vector defining its orientation with respect to other basis functions. The second parameter has the same dimensionality as the input space, and therefore may require a considerable amount of additional storage space when multiple Cloud Segments are present in the basis function.

Parameter	Non-Hierarchical DWRRBF network	Hierarchical DWRRBF network	CBF Network
Best overall recognition rate	84.9%	92.7%	96.1%
Number of basis functions used for best overall recognition rate	44	Level 1 : 42 Level 2 : 17 + 26	9 with 4 Cloud Segments per node
Parameters defining each basis function	1. Discriminative indices 2. Function radius 3. Basis center		1. Discriminative indices 2. Multiple CSRs 3. Basis Center 4. Vectors defining CS orientation
Memory usage	Low	Moderate (due to 3 networks)	High
Computational load	Weighted Distance		1. Weighted Distance 2. Radius Selectivity function

Table 6.9: A summary of operating parameters and performance of DWRRBF and CBF classifiers.

From a neural network point of view a CBF with multiple Cloud Segments can be viewed as a collection of DWRRBFs that are superimposed on each other. All the basis functions in this collection share a single prototype vector as the basis center but have their own function radii for separation from the neighboring basis functions. However, in the CBF, the use of the Radius Selection Function (RSF) (4.29) creates an additional level of non-linearity to the CBF response. Using reference vectors that define the orientation of each Cloud Segment, the RSF further subdivides the responsive region of the basis function in input space into several non-linear partitions. This intra-region partitioning allows the different CSRs of the CBF to be learned using data of their respective partitions and with minimum interference from data belonging to other partitions of the same basis function. Therefore, due to intra-basis partitioning the response of a CBF is based on separate sets of function parameters that are fine-tuned for separating its neighbors based on their direction. In contrast, a DWRRBF node (as well as nodes in traditional RBF networks) responds to all data within the region of the basis function using a single set of function parameters irrespective of the locality of the input. Because of this difference, a CBF becomes more capable of representing the local properties within a region in the input space represented by a single basis function.

6.4 Experiments Using EFR and Half-face Datasets

The two supplementary datasets, the EFR (Expression Feature Regions) dataset and the Half-face dataset described in the previous chapter are of a lower dimensionality compared to the primary dataset used in the experiments described in previous sections. The EFR dataset consists of pixel intensity variations corresponding to three facial regions; the eyes, eye-brows and the mouth that are often accepted as the most important facial regions in the display of facial expressions. Therefore, in addition to the lower dimensionality in the input space, EFR dataset was also expected to be more invariant to differences across different individuals in the image database. The Half-face dataset on the other hand contains only the left-half of images as in the primary dataset and, therefore, provided a 50% reduction in the dimensionality. Moreover, from an anatomical point of view, the human face is regarded as being symmetrical along the vertical plane passing through the tip of the nose [7]. Therefore the left half of a facial image used in the Half-face dataset was assumed to be containing almost all details of the facial expression compared to its full-face counterpart in the primary dataset.

	Accuracy of expression recognition using EFR dataset						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
EFR dataset using 2 level DWRRBF network classifier	66.7%	82.1%	93.2%	70.0%	81.6%	95.4%	83.2%
EFR dataset using non-hierarchical CBF network classifier	72.2%	90.5%	93.2%	75.0%	81.6%	97.7%	86.9%

Table 6.10a: Recognition rates obtained with the EFR dataset.

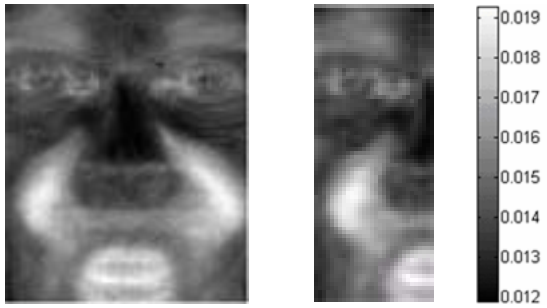
	Accuracy of expression recognition using Half-face dataset						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
Half-Face data using 2 level DWRRBF network classifier	75.8%	93.7%	93.2%	85.0%	87.8%	96.5%	89.8%
Half Face data using non-hierarchical CBF network classifier.	90.9%	96.8%	94.6%	90.0%	89.8%	97.7%	94.2%

Table 6.10b: Recognition rates obtained with the Half-face dataset.

The recognition rates returned by the two datasets for proposed classifiers are presented in Table 6.10a and Table 6.10b, respectively. Contrary to initial expectations results obtained for the EFR dataset showed a lower recognition rate whereas the Half-face dataset performed almost as well as the primary dataset. When compared with primary dataset, (Table 6.2) EFR results showed a drop in recognition rates for Surprise and Disgust expressions while the other expression classes showed an increase in the recognition rate. A closer look at the concentrations of high-valued Discriminative Indices attributed the unexpected performance of the EFR results to the absence of inner-cheek region in the dataset. A majority of the images in source image database had naso-labial folds in the inner cheek region during the display of facial expressions which were prominently captured by the respective Discriminative Indices in primary and Half-face datasets (Figure 6.11a and Figure 6.11b). The absence of these features in the EFR dataset contributed to the lower recognition rates of Surprise and Disgust expressions.

Compared to EFR, dataset performance on the Half-face dataset was more compatible with results obtained earlier with the primary dataset. The best overall recognition rate of 94.2% on the Half-face dataset was slightly lower than that (96.1%) obtained for the Primary dataset. Furthermore, a closer examination of additional misclassifications in the Half-face

dataset revealed that some of the images had slight pose variations. This had affected the symmetry properties, leading to an error in registration of the respective Half-face images. This error in the image registration was identified as the main contributing factor in the lower recognition rate on Half-face dataset compared to the results of the primary dataset.



(a) (b)

Figure 6.11: Example of discriminative indices showing the dominant region of values in the inner cheek / nasal regions. (a) for primary dataset and (b) for Half-face dataset.

6.5 Results Using Other Types of RBF Networks

During several years of development in RBF network related techniques, many enhancements have been suggested to improve their performance as pattern classifiers. Of these the commonly applicable improvements from a high-dimensional classification standpoint were discussed earlier on detail in Chapter 3. In order to compare and contrast the performance of the proposed algorithms against these improved RBF networks some comparative experiments using other types of RBF networks were carried using the Primary dataset. The different types classifiers used in these experiments consisted mainly of RBF networks with their basis functions created according to the following criteria.

a. Gaussian like hyper-spherical basis functions with a function radius (σ) computed according to Euclidean distance criterion. (3.11).

b. Gaussian (hyper-elliptic) basis functions with a diagonal covariance matrix where the diagonal matrix consisted of class-conditional variances for each of the individual features.

c. Gaussian basis functions with a pooled full covariance matrix. The pooled covariance matrix was computed using the entire training dataset and the SVD algorithm (3.48) was used to avoid singularity problems.

d. Gaussian basis functions with their class-conditional full covariance matrix. The SVD algorithm (3.48) was used to avoid the singularity problem of the respective covariance matrices.

The confusion matrices returned using the above types of RBF networks are presented in Tables 6.11a to 6.11d followed by a summary of the recognition rates in all four types in Table 6.12.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**54.5%**	9.1%	9.1%	-	4.5%	22.7%
	Surprise	3.2%	**86.3%**	8.4%	1.0%	-	1.0%
	Sad	5.4%	1.3%	**78.4%**	5.4%	5.4%	4.0%
	Anger	12.5%	7.5%	2.5%	**70.0%**	-	7.5%
	Disgust	6.1%	4.0%	6.1%	2.0%	**71.4%**	10.2%
	Happy	6.9%	2.3%	-	1.1%	1.1%	**88.5%**

Table 6.11a: Confusion matrix for classification using RBF network having Gaussian basis functions with Euclidean radius.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**56.1**%	3.0%	12.1%	1.5%	7.6%	19.7%
	Surprise	1.0%	**87.4**%	5.3%	-	2.1%	4.2%
	Sad	4.0%	1.3%	**83.8**%	2.7%	2.7%	5.4%
	Anger	7.5%	5.0%	12.5%	**62.5**%	-	12.5%
	Disgust	8.2%	-	-	2.0%	**89.8**%	-
	Happy	4.6%	-	-	-	-	**95.4**%

Table 6.11b: Confusion matrix for classification using RBF network having Gaussian basis functions with diagonal covariance matrix.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**81.8**%	1.5%	4.5%	1.5%	3.0%	7.6%
	Surprise	1.0%	**97.9**%	-	-	-	1.0%
	Sad	4.0%	1.3%	**83.8**%	6.8%	1.3%	2.7%
	Anger	5.0%	7.5%	15.0%	**70.0**%	-	2.5%
	Disgust	16.3%	-	-	4.1%	**77.5**%	2.0%
	Happy	5.7%	1.1%	1.1%	2.3%	3.4%	**86.2**%

Table 6.11c: Confusion matrix for classification using RBF network having Gaussian basis functions with pooled full covariance matrix.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**78.8**%	1.5%	4.5%	1.5%	4.5%	9.1%
	Surprise	1.0%	**96.8**%	-	-	-	2.1%
	Sad	4.0%	2.7%	**81.1**%	8.1%	1.3%	2.7%
	Anger	7.5%	7.5%	17.5%	**65.0**%	-	2.5%
	Disgust	16.3%	-	-	4.1%	**77.5**%	2.0%
	Happy	4.6%	1.1%	1.1%	2.3%	3.4%	**87.4**%

Table 6.11d: Confusion matrix for classification using RBF network having Gaussian basis functions with class conditional full covariance matrices.

	Accuracy of expression recognition in other types of RBF networks						
	Fear	Surprise	Sad	Angry	Disgust	Happy	**Total**
Total number of images	66	95	74	40	49	87	**411**
RBF Network using Euclidean Radius	54.5%	86.3%	78.4%	70.0%	71.4%	88.5%	**76.9%**
RBF networks using a diagonal matrix of variances	56.1%	87.4%	83.8%	62.5%	89.8%	95.4%	**81.3%**
RBF network using pooled Covariance matrix	81.8%	97.9%	83.8%	70.0%	77.5%	86.2%	**85.2%**
RBF network using Full within class Covariance matrix	78.8%	96.8%	81.1%	65.0%	77.5%	87.4%	**83.7%**

Table 6.12: A summary of best recognition rates obtained using other types of RBF networks.

The results shown in Tables 6.11a to 6.11d and Table 6.12 were the best performances obtained after experimenting with different types of learning algorithms and network parameters. Network parameters that were varied during these experiments included criteria for addition of new basis functions and learning strategies for basis function parameters and the post-basis mapping. Traditional basis functions with the Euclidean radius showed their best performance with iterative addition of new basis functions. On the contrary, the other three types of networks performed best when clustering algorithms were used to determine their respective basis centers. For the rest of the network parameters including the function radius (σ) and the post-basis mapping, best results were observed with the use of gradient-descent learning algorithm for all four networks.

From the four different types of RBF networks tested, two classifiers using full covariance matrices in basis functions showed better performance while the lowest performance was observed with the network using basis functions based on the Euclidean distance. Between the two RBF networks with full covariance matrices, the one using respective class-conditional covariance matrices performed slightly worse than the network with the pooled covariance matrix. The higher performance of the latter can be attributed to the higher amount of variations retained by the SVD algorithms on the pooled covariance matrix, compared to its class-conditional counterparts. Among the six classes of expressions, Surprise and Happy in general recorded higher recognition rates contributed by the prominent variations in the mouth region. Furthermore, similar to what was observed in previous experiments, a majority of the confusions in all four networks occurred in relation to the Fear and Happy expressions with images from other classes being misclassified as belonging to these expression-classes.

6.6 Performance of Dimensionality Reduction Methods

A common approach in handling high-dimensional spaces for classification is the use of dimensionality reduction techniques on the high-dimensional input. These techniques

usually project the input onto some low-dimensional and possibly uncorrelated subspace, on which the discrimination can be made using low-dimensional classifiers. When using RBF network-based classifiers, the lowered dimensionality of the projected space will allow the optimal use of Gaussian basis functions with their respective full covariance matrices.

There are two major categories of dimensionality reduction methods that are commonly used in facial image recognition. The first referred to as the Eigenface method [56] uses un-labeled images to determine the projection space whereas the second, the Fisherface method, [64] does the same using labeled images. Details of both these techniques were discussed in detail in Chapter 2. In order to test the performance of these techniques on facial expression recognition and compare them with the proposed algorithms, the following experiments were carried out:

a. Classification with a RBF network after Eigenface method was used for dimensionality reduction. The low dimensional feature space was computed by projecting the input onto an eigenspace spanned by the first 31 principal components, computed over the entire training set. The number of available principal components was restricted by the number of training images available per each expression as illustrated in Table 6.1. The figure of 31 components was determined according to the Angry expression, which had only 32 images for each training cycle.

b. Using the same procedure as above, except that the first two principal eigenvectors were excluded in the projection. Consequently the low-dimensional feature space consisted of projections onto the next 29 principal components.

c. Classification with a RBF network after dimensionality reduction by the Fisherface method. Due to the limited number of training samples, the projection matrix was computed according to (2.11) that used a combination of both PCA and Fisher's criterion.

The RBF networks used in all three experiments consisted of Gaussian basis functions with their full class-conditional covariance matrices, computed using the images of respective expression classes. Decision for the selection of 31 principal components in Eigenface approach was made according to the minimum number of 32 images in training sets of Angry expression class.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**81.8**%	1.5%	4.5%	1.5%	3.0%	7.6%
	Surprise	1.0%	**96.8**%	-	-	1.0%	1.0%
	Sad	4.0%	1.3%	**82.4**%	8.1%	1.3%	2.7%
	Anger	5.0%	7.5%	17.5%	**65.0**%	2.5%	2.5%
	Disgust	16.3%	-	2.0%	4.1%	**75.5**%	2.0%
	Happy	5.7%	1.1%	1.1%	2.3%	3.4%	**86.2**%

Table 6.13a: Confusion matrix for classification after dimensionality reduction with Eigenface method.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**84.8**%	-	4.5%	1.5%	3.0%	6.1%
	Surprise	1.0%	**97.9**%	-	-	-	1.0%
	Sad	4.0%	1.3%	**83.8**%	6.8%	1.3%	2.7%
	Anger	5.0%	7.5%	15.0%	**70.0**%	-	2.5%
	Disgust	16.3%	-	-	4.1%	**81.6**%	6.1%
	Happy	4.6%	1.1%	1.1%	2.3%	1.1%	**88.5**%

Table 6.13b: Confusion matrix for classification after dimensionality reduction with Eigenface method with first two principal components removed.

		Network output					
		Fear	Surprise	Sad	Anger	Disgust	Happy
Class label	Fear	**90.9**%	1.5%	-	1.5%	1.5%	4.5%
	Surprise	1.0%	**96.8**%	-	-	1.0%	1.0%
	Sad	2.7%	1.3%	**89.2**%	2.7%	1.3%	2.7%
	Anger	5.0%	5.0%	7.5%	**80.0**%	-	2.5%
	Disgust	12.2%	-	-	4.1%	**81.6**%	2.0%
	Happy	3.4%	1.1%	-	2.3%	2.3%	**90.8**%

Table 6.13c: Confusion matrix for classification after dimensionality reduction with Fisherface method.

The results, in terms of respective confusion matrices obtained in these experiments, are presented in Tables 6.13a to 6.13c followed by a summary of the overall recognition rates of the three classifiers in Table 6.14.

	Accuracy of expression recognition in using dimensionality reduction methods						
	Fear	Surprise	Sad	Angry	Disgust	Happy	Total
Total number of images	66	95	74	40	49	87	411
Eigenface method with 31 components	81.8%	96.8%	82.4%	65.0%	75.5%	86.2%	83.9%
Eigenface method (first 2 components removed)	84.8%	97.9%	83.8%	70.0%	81.6%	88.5%	86.6%
Fisherface method	90.9%	96.8%	89.2%	80.0%	81.6%	90.8%	89.8%

Table 6.14: A summary of recognition rates obtained with RBF networks after dimensionality reduction of input by various techniques.

The overall results showed a general increase in recognition rates with classifiers that use dimensionality reduction techniques compared to those using other types of RBF networks in the high-dimensional input space. The best recognition rates were obtained for the Fisherface method. Compared to the Eigenface method, in the Fisherface approach, the projection matrix is computed using labeled image samples in order to maximize their separation on the projected space. Consequently, the technique retains most of the

information that is relevant to the subsequent discrimination of classes in comparison to the Eigenface method which attempts to retain the larger variations regardless of their relevance to the discrimination.

After several experiments using the Eigenface approach, it was observed that the overall recognition rate peaked when the first two principal components were removed from the projection matrix. Previously, Belhumeur et. al. [64] suggested that the first three components of the Eigenspace corresponded to variations originating from lighting conditions in their experiment. Therefore removing first three eigenvectors from the projection matrix would be likely to eliminate most of the variations due to lighting changes and shadows. However, with respect to images used in these experiments this is not necessary because all images were normalized for variations in intensity prior to creation of their respective datasets. Furthermore, the presence of shadows was almost negligible in the source images that were used in the dataset. As a result, the net increase in recognition rate was attributed more to the removal of some of the significant but irrelevant variations that were captured by the first and the second eigenvectors in the Eigenface approach. On the other hand, due to the use of class specific projections, the problem of these irrelevant variations was not present in the Fisherface approach.

6.7 Comparison of Proposed Classifiers with Other RBFN Based Methods for Holistic Recognition of Facial Expressions

From the experimental results it can be seen that both types of proposed classifiers here have outperformed all other RBF network-based classifiers in terms of their overall recognition rates. The best performance came from the new CBF network which showed an overall recognition rate of 96.1%. The hierarchical DWRRBF network on the other hand was able to correctly recognize 92.7% of the facial expressions in the same dataset. Both these results were superior to the best performances recorded with the other types of RBF networks.

Among other categories of classifiers, Fisherface method with RBF classifier recorded the best performance with an overall recognition rate of 89.8%.

Holistic recognition of facial expressions requires classifiers with different properties compared with those used for their feature-based counterparts. The problem domain of the former is characterized by high-dimensional input spaces that contain a significant amount of irrelevant information. Most of this irrelevant information in holistic approaches originates from structural differences in the faces of different people and causes interference in the discrimination of their facial expressions. Consequently, classifiers used in these systems must have a higher capability in extracting relevant variations from the noisy input while, at the same time, being insensitive to the irrelevant ones.

The experiments described this chapter yield two significant observations. First an increase in the recognition rate was noticeable in the Eigenface approach when first two principal components were discarded (Table 6.14). Second, the Fisherface approach which attempts to optimize projections in the directions of higher separability according to the specified class labels returned recognition rates that were significantly higher than the non-class specific Eigenface approach. Both of these observations point to the same conclusion: that subject-dependant variations are present in the dataset which are of less relevance for discriminating facial expressions.

The lower performance of networks that used the full covariance matrices of high dimensionality can be attributed to the lack of training samples, so that the SVD algorithm computes the inverse of covariance matrix based on the number of non-zero eigenvalues present in the sample covariance matrix. As a result, with N distinct training samples, the maximum extent of information captured by SVD algorithm is limited to variations in $N-1$ principal directions of the input space. However, as mentioned earlier some of these largest

variations are unlikely to support the discrimination of expression classes, and will interfere with the Gaussian boundaries of basis functions for the reasons discussed in Chapter 4, thereby affecting generalization and recognition accuracy of the classifier.

The above problem is somewhat less prominent in the pooled covariance matrix compared to the class-conditional covariance matrices. Since the pooled covariance is computed using all data in the training set, the matrix has a higher rank compared to class-conditional covariance matrices computed using samples of the individual pattern classes. Consequently, the SVD algorithm on the pooled matrix is able to retain more directions of variations present in the training set. On the other hand, use of a single covariance matrix for all basis functions is likely to affect the performance of some of the pattern classes that are separated mainly by locally important variables. For instance, the principal axes of the pooled covariance matrix will be determined by major variations that are present in majority of input data and therefore could be different from features that are important for the local separation of these basis functions.

The above observations can also be used to explain the compatibility of recognition rates that were observed in networks using the full covariance matrices and those using the Eigenface-based dimensionality reductions. Similar to SVD algorithm, the principal components used in Eigenface approach retain the significant variations in input space regardless of their applicability for subsequent discrimination. As a result both methods demonstrated compatible recognition patterns (Table 6.12 and Table 6.14) in spite of the further reduction of dimensionality in the SVD (Eigenface)-based method. The Fisherface approach on the other hand was able to eliminate some irrelevant variations in its projected space and therefore showed a higher accuracy compared to the Eigenface/SVD approach.

Another property of the holistic input that affects conventional RBF networks is the fact that natural clusters in input may not necessarily follow the class structure of facial expressions.

Often variations in facial expressions are smaller than variations caused by some other characteristics of the input. For instance more prominent and larger grouping may occur due to variations in ethnicity or gender of the test subjects in a training image set. The use of supervised clustering on the other hand would lead to multimodal and possibly overlapping data clusters that, as a result, lower the separability of basis space and require greater number of basis functions for accurate representation. Further subdivision of these clusters according to the natural data distribution is also unlikely to be effective due to other reasons. For instance, a subdivision will further reduce the number of training samples available for each cluster in the new structure, thereby causing further difficulties in the estimation of their parameters.

The proposed techniques overcome these issues of the classification problem through novel approaches. These include the introduction of new types of basis functions, namely the Differentially Weighted Radius Radial Basis Function (DWRRBF) and the Cloud Basis Functions (CBF). The former uses differential scaling of distance metric to emphasize the differences that are important for the local separation of the basis function. The latter adds another level of non-linearity to the basis function by sub-partitioning the basis functions according to their local properties. It must also be noted that the purpose of the Discriminative Indices used in the above is different from the use of "Feature Weights" that have been proposed recently for RBF networks [121]. In contrast to Discriminative Indices, feature weights operate on input features directly and are initialized and learned independently of the parameters of basis functions, using approaches similar to learning of the weights in the post-basis mapping. In comparison, Discriminative Indices used in the proposed algorithms are initialized according to the class structure of the problem and thereafter are further fine-tuned in conjunction with the other parameters of the basis function.

6.8 Summary

In this chapter, detailed results produced by a number of classification experiments using datasets described in the previous chapter were discussed. The experiments included those using the new classifiers proposed in this thesis as well as those using some of the traditional RBF network-based classifiers. Results obtained from these experiments showed that both of the proposed algorithms (DWRRBF network and CBF network) outperformed all other types of RBF networks for facial expression classification. The best performance of 96.1% was observed for the CBF network using the primary dataset. The DWRRBF network yielded a recognition rate of 92.7% using a two-level hierarchical classification system. Among the other types of RBF network classifiers, those using Fisherface criterion as the dimensionality reduction method performed best with an overall recognition rate of 89.8%. The lowest performance of 76.9% was recorded for a conventional RBF network with spherical basis functions.

The lower performance of the conventional types of RBF networks was attributed to their inability to effectively deal with irrelevant variations; mainly due to subject's identity information in input space and the inefficiency of these methods in separating such information from those useful for the discrimination of facial expressions. Majority of the traditional approaches base their discrimination on larger variations in the input and therefore were affected by the irrelevant variations in learning a general mapping for the problem domain. The Fisherface criterion on the other hand uses class-specific projections and was more capable in handling these variations compared to the other dimensionality reduction methods. However, the effectiveness of the Fisherface method too was limited by the singularity problem caused by the lack of a large training data set. The proposed classifiers on the other hand were able to handle these conditions more effectively using the novel

approach described in Chapter 4, thereby producing recognition accuracies that surpassed all the other methods.

CHAPTER 7

Conclusions and Directions for Future Research

In this thesis, a novel classification system for holistic recognition of facial expressions from static facial images was presented. Two new types of basis functions, namely, the Differentially Weighted Radius Radial Basis Function (DWRRBF) and the Cloud Basis Function (CBF) were introduced for high dimensional classification based on Radial Basis Function (RBF) networks. The new basis functions were optimized for the properties of the input space. The example considered in this thesis was holistic facial expressions. Furthermore, an iterative training algorithm with gradient descent learning of network parameters was also proposed in order to determine parameters of the new classifiers using relatively small training sets. Test results showed that the proposed classifiers were superior to other types of RBF network based classifiers (with and without input dimensionality reduction) in the recognition of facial expressions from a test image database. The best performance of 96.1% overall recognition rate for six classes of universal facial expressions was obtained with the proposed CBF network classifier, while the proposed DWRRBF network classifier in a 2-level hierarchy yielded a recognition rate of 92.7% on the same test dataset. Both these performances were significantly better than the best recognition rate of 89.8% that was obtained from other types of classifiers using the same set of image data.

A graphical summary of the overall performances of the proposed classifiers and other types of classification systems that were tested in this research is given in Figure 7.1. A complete graphical illustration of recognition rates in individual classes of facial expressions is available in Appendix A.

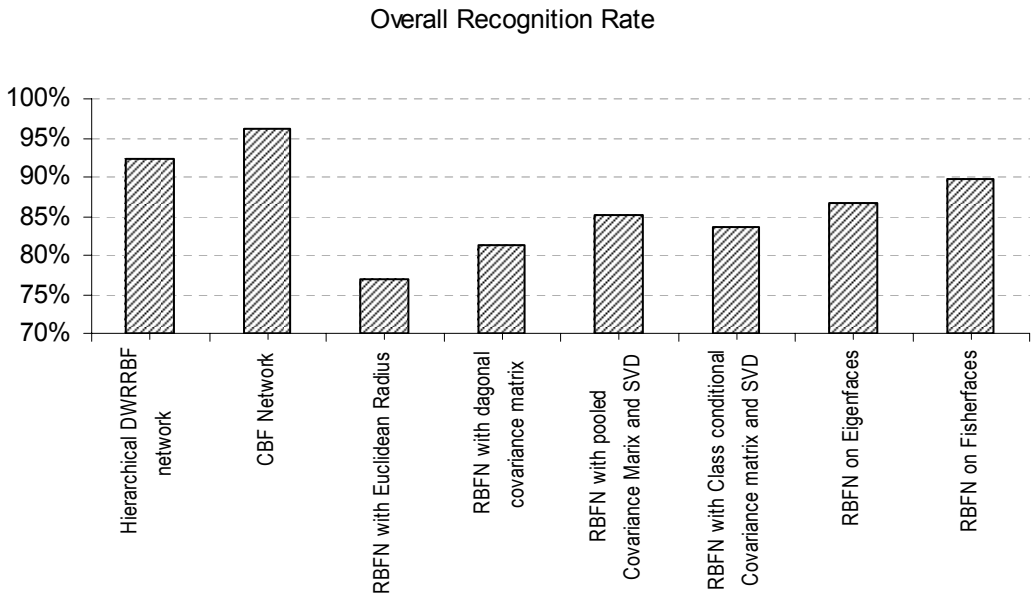


Figure 7.1: A summary of overall performance of different types of classification systems using test image database.

It was found from the results that in general, dimensionality reduction methods based on class-specific projections (i.e. Fisherface method) performed better than those based on maximum variance projections (i.e. Eigenface method) in recognition of all classes of facial expressions. Furthermore, an increase in performance was observed in the latter case when the first two principal eigenvectors were removed from the projection matrix. Both these observations support the hypothesis that not all significant variations in holistic image input space are important for the discrimination of facial expressions. From an anatomical point of view these irrelevant variations can be attributed to structural differences in the faces of people from different cultures, age groups and demographic origins etc. Although normalization of input images described in Chapter 5 was expected to remove some of the unwanted variations caused by different facial proportions and registration of facial images, a complete removal of all subject dependant variations is impossible or difficult at best. Information that remained unaffected by the normalization included differences in skin

texture, skin color and variations that were not affected by facial proportions. For example, the differences in overall shape of the lip region between Caucasian and African American subjects were not removed by any of the image normalization procedures.

In spite of their close relationship to Bayes decision theory and their ability represent data distribution in all directions of the input space, RBF networks having Gaussian basis functions with full covariance matrices were found to be less suitable for holistic recognition of facial expressions. This was attributed to the lack of sufficient number of training samples, which in turn required that the inverse of the covariance matrix be approximated by the pseudo inverse computed using the SVD. Therefore, the data represented by the basis functions was restricted to a group of largest variations which however was not necessarily discriminative of facial expressions (due to the presence of large irrelevant variations). The Eigenface approach essentially operates on the same principle, but here, reduced dimension projection weight vectors are used as inputs to the network. This yielded 83.94% accuracy compared to 85.16% with the full pooled covariance matrix and SVD for computing the inverse.

A common alternative to using the full covariance matrix in RBF networks is a diagonal matrix consisting of variances of individual features of the input space. From a network architecture point of view, this approach resembles a DWRRBF network because variance of each feature act as a scale factor for computation of the Euclidean distance along that feature axes. However, experimental results showed that this approach performed worse (81.3%) compared to the DWRRBF network (92.7%). This is attributed to differences in the way that these scale factors are determined. In the former method, the scaling represented only the variability in input space regardless of their discriminative abilities whereas the Raleigh coefficient based scale factors in DWRRBF networks placed more emphasis on variations that were important in the discrimination of facial expression classes. Therefore the latter allowed cluster boundaries of basis functions to be determined according to more relevant

variations, which subsequently resulted in a better separation of expression classes within the basis space.

It was found that between the proposed DWRRBF network and the CBF network, the former was more sensitive to averaging effects. The averaging effect de-emphasized variations that occurred only in some of expression classes and those with low amplitude. In facial images, such variations correspond to expressions of negative emotions and those expressed by facial regions like eye-brows and eyes which have lower amplitude variations compared to those expressed by the mouth region. With the averaging effect the cluster boundaries in DWRRBF networks were less sensitive to these variations, being biased by the more dominant variations and their respective dominant Discriminative Indices. However, this problem was less prominent in the CBF network because of multiple boundary segments used for the separation of basis functions from their neighbors. Multiple boundary segments in the basis function allowed the use of different radii that were determined according to the variability of locally important regions. Consequently, CBF network was capable of representing several decision boundaries in the same basis function and was therefore able to deliver better performance using a relatively smaller basis space compared to its DWRRBF network counterpart having a single radius.

7.1 Directions for Future Research

Although the proposed classifiers were developed for holistic recognition of facial expressions, their capabilities are not restricted to this specific application. Instead, they are likely to show better results in many high-dimensional classification problems that require the discrimination to be made under presence of significant but irrelevant variations in the input. There are many interesting applications which have these properties, especially in areas related to holistic image recognition. For instance in the domain of face processing, these techniques can be applied for classification of gender [122], age groups, ethnic groups

etc. despite subject differences, facial makeup and facial expressions. Furthermore, some initial investigations done during this research have shown indications that the proposed CBF network could perform extremely well in OCR applications [123], especially in holistic recognition of handwritten digits [124][125].

The scope of this research was restricted to investigating the classification aspects of a holistic facial expression recognition system. However the proposed techniques may also be used to search for facial features like the eyes and nose-tip, which are required for the normalization of input. Future research could investigate this idea. It will be highly interesting to build a complete neural network based facial expression recognition system that could use un-processed raw images as the input.

In addition to their higher classification accuracy, the proposed classifiers are less demanding in computational power requirements compared to their counterparts using full covariance matrices or projection based dimensionality reduction methods. Both of the latter techniques require matrix operations in high dimensions whereas in the proposed classifiers the complexity is limited to vector manipulations. Therefore, parallel hardware architectures like array processors and single instruction multiple data (SIMD) architectures would be able to exploit the parallel properties of the new basis functions for maximum throughput compared to other types using matrix operations. Moreover, the layered network topology with relatively simple processing units will allow an efficient implementation using low cost fine-grain Field Programmable Gate Arrays (FPGA) based platforms. Thus an investigation on the implementation of proposed algorithms on a dedicated hardware platform will benefit a number of new embedded applications that require the analysis of human facial expression and emotions.

REFERENCES

[1] A Mehrabian, "Communication without words", *Psychology Today*, vol. 2. no. 4, pp.53-56, 1968.

[2] http://www.sony.net/Products/aibo/aiboflash.html

[3] J. Bulwer, A dissection of the significance muscles of the affections of the mind, London: Humphrey and Moseley, 1949.

[4] G. B. Duchenne de Boulogne, The mechanism of human facial expression, Transl. R. Andrew Cuthbertson, Cambridge University Press, 1990.

[5] P. Ekman, "The argument and evidence about universals in facial expressions of emotions", *Handbook of Social Psychophysiology*, H. Wagner and A Monstead, Ed. John Wiley, pp. 143-146, 1989.

[6] G. Faigin, The Artist's Complete Guide to Facial Expressions, Watson-Guptill, 1990.

[7] F. I. Parke, K. Waters, Computer Facial Animation, A K Peters, 1996.

[8] Ekman P, "Methods for measuring facial action," in *Handbook of Methods in Nonverbal Behaviour Research*, K. Scherer and P. Ekman, Ed. Cambridge University Press, pp. 45-135, 1982.

[9] P. Ekman, and W. Friesen, "Measuring facial movement", *Journal of Environmental Psychology and Nonverbal Behavior,* vol. 1, no. 1, pp. 56-75, 1976.

[10] E. Smith, M. S. Bartlett and J. Movellan, "Computer recognition of facial actions: A study of co-articulation effects", *in 8th Joint Symposium on Neural Computation*, pp. 137-141, California, 2001.

[11] M. Pantic, L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions", *Image and Vision Computing,* vol. 18, no. 1, pp. 881-905, 2000.

[12] C.E., lzard, R.R. Huebner, D. Risser, G.C. McGinnes and L.M. Dougherty, "The young infant's ability to produce discrete emotional expressions", *Developmental psychology*, vol. 16, no. 1, pp.132-140, 1980.

[13] C. H. Hjortsj, Man's Face and Mimic Language, Sweden: Lund, 1970.

[14] C.L. Lisetti, "Motives for intelligent agents: Computational scripts for emotion concepts", *Proceedings of the Sixth International Conference on Artificial Intelligence (SCAI'97),* pp. 59-70, Netherlands, 1997.

[15] P. Ekman, W. Friesen, and M. O'Sullivan, "Smiles when lying", *Journal of Personality and Social Psychology*, vol. 54. vol. 2, pp.414-420, 1988.

[16] P. Ekman, R. Davidson, and W. Friesen, "The Duchenne smile: Emotional expression and brain physiology II", *Journal of Personality and Social Psychology,* vol. 56, no. 2, pp. 342-353, 1990.

[17] L.C. De Silva, T. Miyasato, and F. Kishino, "Emotion enhanced multimedia meetings using the concept of virtual space teleconferencing", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems 96 (ICMC96)*, pp. 28-33, Japan, 1996.

[18] L.C. DE Silva, T. Miyasato, and F. Kishino, "Emotion Enhanced Face to Face Meeting Using the Concept of Virtual Space Teleconferencing", *IEICE Transactions on Information and Systems*, vol. E79-D, no.6, pp. 772-780, 1996.

[19] S. R. Lehky, "Fine discrimination of faces can be performed rapidly", *Journal of Cognitive Neuroscience*, vol. 12. no.5, pp. 848-855, 2000.

[20] Simon Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall, 1999.

[21] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A.J. Smola, and K.-R. Müller, "Invariant Feature extraction and classification in kernel spaces" *Advances in Neural Information Processing Systems 12,* MIT Press, pp. 526-532, 2000.

[22] M. Suwa, N. Sigie, and K. Fujimora, "A Preliminary Note on Pattern Recognition of Human Emotional Expression", *Proceeding of the 4th International Joint Conference on Pattern Recognition*, pp. 408-410, Japan, 1978.

[23] A. Samal, and P.A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A Survey", *Pattern Recognition,* vol. 25. no. 1, pp. 65-77, 1992.

[24] T. Valentin, H. Abdi, A. O'Toole and G. Cottrell, "Connectionist models of face processing: A survey", *Pattern Recognition*, vol. 27, no. 9, pp.1209 – 1230, 1994.

[25] A. W. Yong and H.D. Ellis, Handbook of research on face processing, Elsevier, 1989.

[26] J.N. Bassili, "Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face", *Journal of Personality and Social Psychology*, vol.37, pp. 2049-2059, 1979.

[27] B. Horn and B. Schunck, "Determining optical flow", *Artificial Intelligence*, vol. 17, no. 1, pp. 185-203, 1981.

[28] H. H. Naigel, "On the Estimation estimation of optical flow: Relations between Different Approaches and Some New Results", *Artificial Intelligence,* vol. 33, no. 3, pp. 299-324, 1987.

[29] K. Mase, and A. Pentland, "Lip Reading by Optical Flow," *IEICE Transactions*, vol. J73-D-II, no. 6, pp.796-803, 1990.

[30] K. Mase, "Recognition of facial expression from optical flow", *IEICE Transactions, Special Issue on Computer Vision and its Applications*, vol. E74, no. 10, pp.3474-3483, 1991.

[31] Y. Yacoob and L. Davis, "Recognizing Facial Expression by Spatio-Temporal Analysis", *Proceedings of the 12th International Conference on Pattern Recognition*, pp.747-749, Israel,1994.

[32] Y. Yacoob and L. Davis, "Computing spatio-temporal representation of human faces", in *Proceedings of Computer Vision and Pattern Recognition (CVPR),* pp.70-75, Seattle, WA, 1994.

[33] M. Rosenblum, Y. Yacoob and L.S. Davis, "Human Expression Recognition from Motion Using a Radial Basis Function Network Architecture", *IEEE Transactions on Neural Network,* vol. 7, no. 5, pp.1121-1138, 1996.

[34] M. Yoneyama, Y. Iwano, A. Ohtake, and K. Shirai, "Facial Expression Recognition using Discrete Hopfield Neural Networks", *Proceedings of the International Conference on Image Processing (ICIP*), vol. 3, no. 1, pp.117-120, 1997.

[35] M.J. Black, and Y. Yacoob, "Recognizing facial expressions under rigid and non-rigid facial motions", *Proceedings of the IEEE International Workshop on Automatic Face and Gesture Recognition*, pp.12-17, Zurich, 1995.

[36] I.A. Essa and A.P. Pentland, "Coding, Analysis, Interpretation and Recognition of Facial Expressions", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 7. pp. 757-763, 1997.

[37] Y. Moses, D. Reynard and A. Blake, "Determining Facial Expressions in Real Time," in *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp.332-337, 1995.

[38] B.D.O. Anderson and J.B. Moore, "Optical Filtering", *Electrical Engineering Series,* Prentice Hall, 1979.

[39] T. Otsuka and J. Ohya, "Extracting Facial Motion Parameters by Tracking Feature Points", *Lecture Notes in Computer Science*, Springer-Verlag vol. 1554, pp. 433-444, 1999.

[40] J. Shi, and C. Tomasi, "Good features to track", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.593-600, Seattle,1994.

[41] M. Wang, Y. Iwai and M. Yachida, "Expression Recognition from Time-Sequential Facial Images by use of Expression Change Model", *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, pp.324-329, 1998.

[42] F. Bourel, C.C. Chibelushi, and A.A. Low, "Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localised Facial Dynamics", *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, pp. 113-119, 2002.

[43] T. Bailey and A. K. Jain, "A Note on Distance-Weighted k-Nearest Neighbor Rules", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 1, pp.311-313, 1978.

[44] Y. Tian, T. Kanade and J. F. Cohn, "Recognizing Action Units for facial Expression Analysis", *Technical Report CMU-RI-TR-99-40, Robotics Institute, Carnegie Mellon University,* , December 1999.

[45] J. J. Lien, "Automatic Recognition of Facial Expressions using Hidden Markov Models and Estimation of Expression Intensity", *Ph.D Thesis, The Robotic Institute, Carnegie Mellon University*, 1998.

[46] H. Kobayashi and F. Hara, "Recognition of Six Basic Facial Expressions and Their Strength by Neural Network", *Proceedings of the International Workshop on Robot and Human Communication*, pp. 381-386, 1992.

[47] H. Kobayashi, A. Tange and F. Hara, "Real-Time Recognition of Six Basic Facial Expressions", *Proceedings of the International Workshop on Robot and Human Communication*, pp. 179 – 186, Japan, 1995.

[48] H. Ushida, T. Takagi, T. Yamaguchi, "Recognition of facial expressions using conceptual fuzzy sets", *Proceedings of the Second IEEE International Conference on Fuzzy Systems,* vol. 1, pp.594 -599, 1993.

[49] T. Kohonen, "The neural phonetic typewriter," *IEEE Computer,* vol.21, no.3, pp. 11-22, 1988.

[50] J. Chang; J. Chen, "A facial expression recognition system using neural networks", *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99),* vol. 5, pp. 3511-3516, Washington DC, 1999.

[51] C. L. Huang and C. W. Chen, "Human facial feature extraction for face interpretation and recognition", *Pattern Recognition,* vol. 25, no.12, pp.1435-1444, 1992.

[52] G.D. Kearney, S. McKenzie, "Machine Interpretation of Emotion: Design of a Memory-Based Expert System for Interpreting Facial Expressions in Terms of Signaled Emotions (JANUS)", *Cognitive Science,* vol.17, no. 2, pp. 589-622, 1993.

[53] M. Pantic, "Human Emotion Recognition Clips Utilized Expert System: HERCULES", *Master Thesis, KBS Group, TU Delft, Netherlands*, 1996.

[54] M. Pantic and L. Rothkrantz, "Automatic Recognition of Facial Expressions and Human Emotions", *Proceedings of ASCI 97*, pp. 196-202, Netherlands, 1997.

[55] A. Samal, "Minimum resolution for human face detection and identification", *SPIE Human Vision, Visual Processing, and Digital Display II,* vol. 1453, pp. 81-89, 1991.

[56] M.Turk and A.Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71- 86, 1991.

[57] L. Sirivich and M. Birby., "Low dimensional procedure for the characterization of human faces", *Journal of Optical Society of America*, vol. A4, no. 3, pp. 519-524, 1987.

[58] L. Daw-Tung, J. Chen, "Facial Expressions Classification with Hierarchical Radial Basis Function Networks", *Proceedings of 6th International Conference on Neural Information Processing, ICONIP '99*, vol. 3, pp. 1202-1207, 1999.

[59] C. Padgett, G. Cottrell and R. Adolphs, "Categorical perception in facial emotion classification", *Proceedings of the 18th Annual Cognitive Science Conference*, pp.249-253, San Diego, 1996.

[60] C. Padgett and G.W. Cottrell, "A simple neural network models categorical perception of facial expressions", *Proceedings of the Twentieth Annual Cognitive Science Conference*, pp.806-807, New Jersey, 1998.

[61] C. Padgett and G. Cotrell, "Identifying emotion in static face images", *Proceedings of the 2nd Joint Symposium on Neural Computation,* vol. 5, pp.91-101, California, 1995.

[62] M. Dailey, G. Cottrell and R. Adolphs, "A six-unit network is all you need to discover happiness", *Proceedings of the Twenty-Second Annual Cognitive Science Conference*, pp. 1210 – 1215, New Jersey, 2000.

[63] J.G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters", *Journal of Optical Society America,* vol.2, no. 3, pp. 1160-1169, 1985.

[64] P. N. Belhumeur, J.P. Hespanha, D.J. Kriegman, **"**Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711 -720, 1997.

[65] R. Duda and P. Hart, Pattern Classification and Scene Analysis, New York: Wiley. 1973.

[66] E. T. Rolls and A. Treves, Neural Networks and Brain Function. Oxford, 1998.

[67] L. Franco, and A. Treves, "A Neural Network Face Expression Recognition System using an Unsupervised Local Processing", *Proceedings of the Second International Symposium on Image and Signal Processing and Analysis (ISPA'01),* pp. 628-632, Croatia, 2001.

[68] A. Katoh, Y. Fukui, "Classification of facial expressions using self-organizing maps", *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* vol. 2, pp. 986 -989, 1998.

[69] C. L. Lisetti and D. E. Rumelhart, "Facial expression recognition using a neural network", *Proceedings of the 11th International Florida Artificial Intelligence Research Society Conference (FLAIRS'98)*, pp. 328-332, Menlo Park, CA, 1998.

[70] Y. Inooka, M. Fukumi, N. Akamatsu, "Learning and Analysis of Facial Expression Images Using a Five-Layered Hourglass-Type Neural Network", *Proceedings of IEEE SMC '99 Conference on Systems, Man and Cybernetics,* vol. 5, pp. 373-376, 1999.

[71] A. Colmenarez, B. Frey and T. S. Huang, **"**A Probabilistic Framework for Embedded Face and Facial Expression Recognition", in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1592-1597, Colorado, 1999.

[72] P. Ekman and W.V. Friesen, Facial Action Coding System (FACS) Manual, Palo Alto: Consulting Psychologists Press, 1978.

[73] M. Pantic and J. M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, pp. 1424-1445, 2000.

[74] R. Chellappa, C.L. Wilson, S. Sirohey, "Human and machine recognition of faces: a survey", *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705 -741, 1995.

[75] J. M. Gilbert and W. Yang, "A real-time face recognition system using custom VLSI hardware", *Proceedings of IEEE Workshop on Computer Architecture for Machine Perception,* pp. 58-66, New Orleans, 1993.

[76] C.E. Cox, W.E. Blanz, "GANGLION: A Fast Field-Programmable Gate Array Implementation of a Connectionist Classifier", *Journal of Solid State Circuits*, vol. 27, no. 3, pp. 288-299, 1992.

[77] V. Salapura, M. Gschwind and O. Maischberger, "A Fast FPGA Implementation of a General Purpose Neuron", *Porceedings of the Fourth International Workshop on Field Programmable Logic and Aplications*, pp. 1105-1109, Czech Republic, 1994.

[78] M.J.D. Powel, "Radial basis functions for multivariate interpolation: a review", *Algorithms for Approximation*, J.C. Mason and M.G. Cox Ed. Oxford Clarendon Press. pp. 143-167, 1987.

[79] C.A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions", *Constructive Approximation.* vol.2, pp. 11-22, 1986.

[80] D.S. Broomhead and D. Lowe, "Multivariate functional interpolation and adaptive networks", *Computer Systems*, vol.2, pp. 321-355, 1988.

[81] J.E. Moody and C.J. Darken, "Fast learning in networks of locally-tuned processing units", *Neural Computation*, vol. 1, no. 2, pp.281-294, 1989.

[82] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.

[83] E. J. Hartman, J.D. Keeler and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations", *Neural Computations*, vol. 2, no. 2, pp.210-215, 1990.

[84] J. Park and I.W. Sandberg, "Universal approximation using radial basis function networks", *Neural Computation*, vol. 3, no. 2, pp.246-257, 1991.

[85] J. Park and I.W. Sandberg, "Approximation and radial basis function networks", *Neural Computation*. vol. 5, no. 2, pp.305-316, 1993.

[86] T. Poggio and F. Girosi, "Networks for approximation and learning", *Proceedings of the IEEE,* vol. 78, no. 9, pp.1481-1497, 1990.

[87] T.M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition", *IEEE Transactions on Electronic Computers*, vol. EC-14, pp.326-334, 1965.

[88] R. O. Duda, P. E. Hart and D.G. Stork, Pattern Classification, John Wiley, 2001.

[89] M. Kraijveld and R. Duin. "Generalization capabilities of minimal kernel-based networks", *Proceedings of the IEEE International Joint Conference on Neural Networks,* vol. 1, pp. 843-848, New York, 1991.

[91] K. Fukunaga and R.R. Hayes, "The reduced Parzen classifier", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 4, pp. 423-425, 1989.

[92] H. Demuth and M. Beale, Neural Network Toolbox for use with Matlab, The Mathwotks Inc. 2001.

[93] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification", *International Journal of Control*, vol. 50, no. 5, pp.1873-1896, 1989.

[94] S. Chen, C.F.N. Cowan, and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp.302-309, 1991.

[95] Z. Wang and T. Zhu, "An efficient learning algorithm for improving generalization performance of radial basis function neural networks", *Neural Networks*, vol. 13, no. 4, pp. 545-553, 2000.

[96] R.C. Dubes and A.K. Jain, Algorithms for Clustering Data, Prentice Hall, 1988.

[97] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley & Sons, 1990

[98] Lloyd S.P. "Least squares quantization in PCM", *IEEE Transactions on Information Theory*. vol. 28, no. 2, pp.129-137, 1982.

[99] Y. Linde, A Buzo and R.M. Gray, "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, vol. 28, no. 1, pp.84-95, 1980.

[100] D. Judd, P. McKinley and A. Jain. "Large-Scale Parallel Data Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 871-876, 1998.

[101] J. White, V. Faber and J. Saltzman, United states Patent No. 5,467,110. Nov. 1995.

[102] V. Ramasubramanian and K. Paliwal, "Fast K-Dimensional Tree Algorithms for Nearest Neighbor Search with Applications to Vector Quantization Encoding", *IEEE Transactions on Signal Processing*, vol. 40, no. 3, pp.518-531,1992.

[103] K. Alsabti, S. Ranka and V. Singh, “An Efficient K-means Clustering Algorithm”, *Proceedings of the First Workshop on High-Performance Data Mining*, pp. 106-113, Orlando, Florida, 1998.

[104] J. Bilmes, “A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models”, ICSI-TR-97-021, University of Berkeley Technical report, CA, 1998.

[106] F. Schwenker, H. A. Kestler and G. Palm, “Three learning phases for radial-basis-function networks”, *Neural Networks,* vol. 14, no. 4, pp. 439-458, 2001.

[107] Y. Hwang and S. Bang, “An Efficient Method to Construct a Radial Basis Function Neural Network Classifier”, *Neural Networks,* vol. 10, no. 8, pp.1495-1503, 1997.

[108] F. Belloir, A. Fache, A. Billat, “A general approach to construct RBF Net-Based Classifier”, *Proceedings of the European Symposium on Artificial Neural Networks*, pp. 399-404, Belgium, 1999.

[109] K. I. Diamantaras, S. Y. Kung, Principle component neural networks: theory and applications, John Willey and Sons, 1996.

[110] S. Tadjudin, D.A. Landgrebe, “Covariance estimation with limited training samples**”,** *IEEE Transactions on Geosciences and Remote Sensing*, vol. 37, no. 4, pp. 2113 -2118, 1999.

[111] J.P. Hoffbeck, D.A. Landgrebe, “Covariance matrix estimation and classification with limited training data”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp.763 -767, 1996.

[112] J. Yuan, T.L. Fine, "Neural network design for small training sets of high dimension", *IEEE Transactions on Neural Networks,* vol. 9, no. 2, pp.266-280, 1998.

[113] L.O. Jimenez, D.A. Landgrebe, "Supervised classification in high-dimensional space: geometrical, statistical and asymptotical properties of multivariate data", *IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews*. vol. 28, no. 1, pp. 39-54, 1998.

[114] S.J. Raudys, A.K. Jain, "Small sample size effects in statistical pattern recognition: recommendations for practitioners", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp.252-264, 1991.

[115] M. Stone, "Cross-validity choice and assessment of statistical predictions", *Journal of the Royal Statistical Society*, vol. B36, no. 1, pp. 11-147, 1973.

[116] Yale University, Yale face Database, Available:http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[117] Purdue University, The AR Face Database, Available: http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html

[118] University of Stirling, The Psychological Image Collection at Stirling (PICS), Available:http://pics.psych.stir.ac.uk

[119] Y. Tian, T. Kanade and J.F. Cohn, "Recognizing Action Units for Facial Expression Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 97-115, 2001.

[120] University of Evanville, The Figure Drawing Lab, Available:http://www2.evansville.edu/drawinglab/face.html

[121] T.A. Hoang, D.T. Nguyen, "Optimal learning for pattern classification in RBF networks", *Electronic Letters*, vol. 38, no. 20, pp. 1188-1190, 2002.

[122] R. Brunelli and T. Poggio, "Cultural effects in automated face perception", *Biological Cybernetics*, vol. 69. pp. 235-241, 1993.

[123] S. Mori, C.Y. Suen and K. Yamamoto, "Historical review of OCR research and development", *Proceedings of the IEEE,* vol. 80, pp.1029-1058, 1992.

[124] Y. Lee, "Handwritten digit recognition using K nearest neighbor, radial basis function and back-propagation Neural Networks", *Neural Computation*, vol. 3, pp. 440-449, 1991.

[125] L. Bottou, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, L.D. Jackel, Y. LeCun, U.A. Muller, E. Sackinger, P. Simard, V. Vapnik, "Comparison of Classifier methods; A case study in handwritten digit recognition", *Proceedings of the IEEE International Conference on Pattern Recognition,* vol. 2, pp.77-82, Los Alamitos, CA, 1994.

Appendix A

Recognition accuracies recorded for six universal expression classes Vs Different types of classification systems